Intermediate System Administration for the Solaris™ 9 Operating Environment SA-239

Student Guide

Sun Microsystems, Inc.
UBRM05-104
500 Eldorado Blvd.
Broomfield, CO 80021
U.S.A.

Revision A.2

Export Commodity Classification Number (ECCN) assigned: 12 December 2001

SA239_A2_0903_2_F

Table of Contents

Module 9

Performing Boot and Shutdown Procedures

Objectives

Upon completion of this module, you should be able to:

- Identify run level fundamentals
- Identify the phases of the boot process
- Control boot processes
- Perform system shutdown procedures

The following course map shows how this module fits into the current instructional goal.

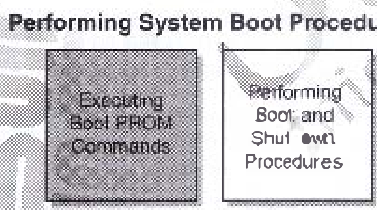

Figure 9-1 Course Map

Identifying Run Level Fundamentals

A run level is a system state, represented by a digit or letter, that defines what services and resources are currently available to users. The system is always running in a single run level.

Solaris OE Run Levels

Table 9-1 shows the eight run levels found in the Solaris OE.

Table 9-1 Solaris OE Run Levels

Run Level	Function
0	System is running the PROM monitor.
s or S	Solaris OE single-user mode with critical file systems mounted and accessible.
1	The system is running in a single-user administrative state with access to all available file systems.
2	The system is supporting multiuser operations. Multiple users can access the system. All system daemons are running except for the Network File System (NFS) server and some other network resource server related daemons.
3	The system is supporting multiuser operations and has NFS resource sharing and other network resource servers available. Specified as the default run level in the `/etc/inittab` file.
4	This level is currently not implemented.
5	A transitional run level in which the Solaris OE is shut down and the system is powered off.
6	A transitional run level in which the Solaris OE is shut down and the system reboots to the default run level.

Determining a System's Current Run Level

To determine the current run level of a system, use the `who -r` command.

Figure 9-2 shows output from the command.

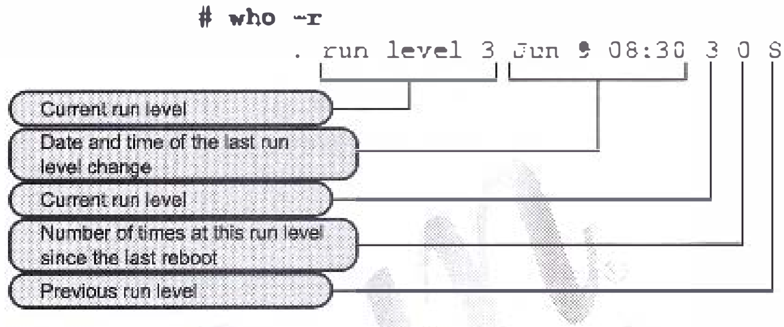

Figure 9-2 The System's Current Run Level

Changing Run Levels

Run levels are sometimes referred to as `init` states because the `init` process transitions between run levels. You can use the `init` command to manually initiate run-level transitions. You can also change run levels with the `shutdown`, `halt`, `reboot`, and `poweroff` commands.

Identifying the Phases of the Boot Process

In general, when a system is powered on, the PROM monitor runs a POST procedure that checks the hardware and memory on the system. If no errors are found, and the auto-boot? parameter is set to true, the system begins the automatic boot process.

The entire boot process is described by four distinct phases:

- The boot PROM phase
- The boot programs phase
- The kernel initialization phase
- The init phase

Figure 9-3 shows the phases of the boot process.

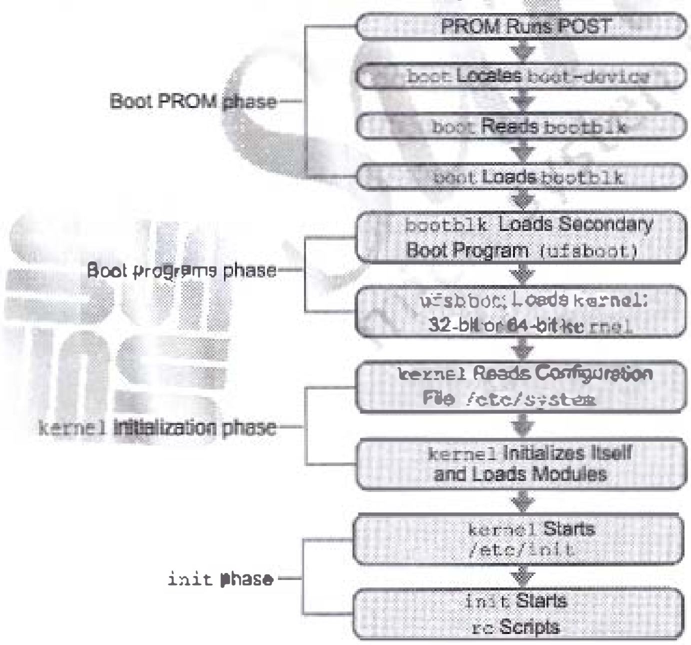

Figure 9-3 Phases of the Boot Process

Boot PROM Phase

The boot PROM performs the following steps during the first part of the boot sequence:

- The PROM runs the POST.

  The boot PROM firmware runs the POST to verify the system's hardware and memory. It then begins its boot sequence upon successful completion of the self-test diagnostics.

- The PROM displays the system identification banner.

  The model type, processor type and speed, keyboard status, PROM revision number, amount of installed random access memory (RAM), NVRAM serial number, Ethernet address, and host ID are displayed.

- The boot PROM determines the boot device by reading the PROM parameter `boot-device`.

- The boot PROM reads the disk label located at Sector 0 on the default boot device.

- The boot PROM finds the `boot` program from the default boot device programmed into the PROM.

  The boot PROM program reads a system's primary boot program called `bootblk` (located at Sectors 1 through 15) that contains a UNIX file system (ufs) file system reader. (The `bootblk` program is placed on the disk by the `installboot` program during system installation.)

  The `boot` command loads the `bootblk` program from its location on the boot device into memory.

Boot Programs Phase

The following describes the boot programs phase:

- The `bootblk` program loads the secondary boot program, `ufsboot`, from the boot device into memory.

  The path to `ufsboot` is recorded in the `bootblk` program, which is installed by the Solaris OE utility `installboot`.

- The `ufsboot` program locates and loads the appropriate two-part `kernel`.

The core of the kernel is two pieces of static code called genunix and unix, where genunix is the platform-independent generic kernel file and unix is the platform-specific kernel file.

When ufsboot loads these two files into memory, they are combined to form the running kernel.

On a system running in 32-bit mode, the two-part kernel is located in the directory /platform/`uname -m`/kernel.

On a system running in 64-bit mode, the two-part kernel is located in the directory /platform/`uname -m`/kernel/sparcv9.

Note – To determine the platform name (for example, the system hardware class), type the uname -m command. For example, when you type this command on an Ultra 10 workstation, the console displays sun4u.

The kernel Initialization Phase

The following describes the kernel initialization phase:

- The kernel reads its configuration file, called /etc/system.
- The kernel initializes itself and begins loading modules.

  The kernel uses the ufsboot command to load the files. When it has loaded enough modules to mount the root file system, it unmaps the ufsboot program and continues.
- The kernel starts the /etc/init process.

Note – The /etc/init and /sbin/init processes are linked together.

The SunOS™ kernel is a small static core, consisting of genunix and unix and many dynamically loadable kernel modules.

Modules can consist of device drivers, binary files to support file systems, and streams, as well as other module types used for specific tasks within the system.

The modules that make up the kernel typically reside in the directories /kernel and /usr/kernel. Platform-dependent modules reside in the /platform/`uname -m`/kernel and /platform/`uname -i`/kernel directories.

Each subdirectory located under these directories (see Figure 9-4) is a collection of similar modules.

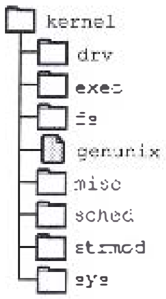

Figure 9-4 Module Subdirectories in the /kernel Directory

The following describes the types of module subdirectories contained in the /kernel, /usr/kernel, /platform/`uname -m`/kernel, or /platform/`uname -i`/kernel directories:

- drv – Device drivers
- exec – Executable file formats
- fs – File system types, for example, ufs, nfs, and proc
- misc – Miscellaneous modules (virtual swap)
- sched – Scheduling classes (process execution scheduling)
- strmod – Streams modules (generalized connection between users and device drivers)
- sys – System calls (defined interfaces for applications to use)

The /kernel/drv directory contains all of the device drivers that are used for system boot. The /usr/kernel/drv directory is used for all other device drivers.

Modules are loaded automatically as needed either at boot time or on demand, if requested by an application. When a module is no longer in use, it might be unloaded on the basis that the memory it uses is needed for another task.

After the boot process is complete, device drivers are loaded when devices, such as tape devices, are accessed. This process is called autoconfiguration because some kernel driver modules are loaded automatically when needed.

Upon initial or reconfiguration boot, the system does a self-test and checks for all devices that are attached.

The advantage of this dynamic `kernel` arrangement is that the overall size of the `kernel` is smaller, which makes more efficient use of memory and allows for simpler modification and tuning. Figure 9-5 shows this arrangement.

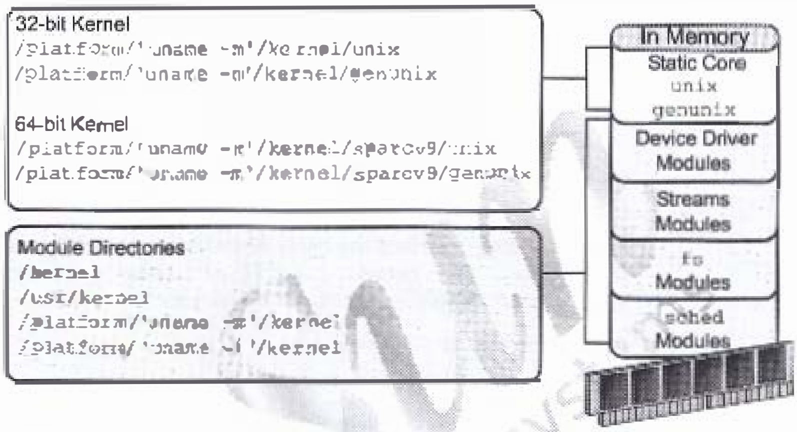

Figure 9-5 The `kernel` and Modules Loaded in Memory

Note – The `sparcv9` CPU is the type of CPU that supports 64-bit processing.

The `/etc/system` File and `kernel` Configuration

Caution – The Solaris OE builds the `kernel` based upon the size of the system (memory, CPUs, and so on). In almost all cases, the performance of the default `kernel` that is built is quite adequate to handle most day to day activities on the system. Any modifications should be made with extreme caution.

The `/etc/system` file is the control file for modifying which modules and parameters are to be loaded by the `kernel` at boot time. By default, all lines in this file are commented out.

Modifying the kernel's behavior (or configuration) requires editing the /etc/system file. Altering this file allows you to modify the kernel's treatment of loadable modules as well as to modify kernel parameters for some performance tuning.

The ufsboot program contains a list of default loadable kernel modules that are loaded at boot time. However, you can override this list by modifying the /etc/system file to control which modules, as well as which parameters, are loaded.

All changes to this file take effect after a reboot.

The /etc/system file can explicitly control:

- The search path for default kernel modules to be loaded at boot time
- The root file system type and device
- The modules that are excluded from loading automatically at boot time
- The modules to be forcibly loaded at boot time, rather that at first access
- The new values to override the default kernel parameter values

Note – Command lines must be 80 characters or less in length, and comment lines must begin with an asterisk (*) and end with a newline character.

The /etc/system file is divided into five distinct sections:

- moddir:

  Sets the search path for default loadable kernel modules. You can list together multiple directories to search, delimited either by blank spaces or colons. If the module is not found in the first directory, the second directory is searched, and so on.

- `rootdev` and `rootfs` system configuration:

  Sets the root file system type to the listed value. The default is `rootfs:ufs`.

  Sets the root device. The default is the physical path name of the device on which the boot program resides. The physical path name is platform dependent and configuration dependent. The following is an example path:

  `rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a`

- `exclude`:

  Does not allow the loadable kernel modules to be loaded during kernel initialization, for example:

  `exclude: sys/shmsys`

- `forceload`:

  Forces the kernel modules to be loaded during kernel initialization, for example:

  `forceload: drv/vx`

  The default action is to load a kernel module automatically when its services are first accessed during runtime by a user or an application.

- `set`:

  Changes kernel parameters to modify the operation of the system, for example:

  `set maxusers=40`

Editing the /etc/system File

Before you edit the /etc/system file, you should make a backup copy. If you enter incorrect values in this file, the system might not be able to boot.

The following example shows how to copy the original /etc/system file to a backup file and then edit the /etc/system file.

```
# cp /etc/system /etc/system.orig
# vi /etc/system
```

If a boot process fails because of an unusable /etc/system file, issue the interactive boot command: `boot -a`. When you are requested to enter the name of the system file, type in the name of your backup system file, or, alternatively, enter /dev/null for a null configuration file.

The `init` Phase

The final phase of the boot process is the `init` phase. During this phase, the `init` daemon starts the run control (`rc`) scripts that start other processes. The `init` daemon is a general process spawner. Its primary role is to create processes from information stored in the file `/etc/inittab`.

The `init` daemon executes system start up (`rc`) scripts that, in turn, execute a series of other scripts.

After the `init` phase completes successfully, the default behavior is to display the system command-line login prompt or the GUI login window.

The `/etc/inittab` File

When you boot a system or change run levels with the `init` or `shutdown` command, the init daemon stops process, starts processes, or does both by reading information from the `/etc/inittab` file.

The `inittab` file defines three important items for the `init` process:

- The system's default run level
- What actions to take when the system enters a new run level
- What processes to start, monitor, or restart if terminated

Each line entry in the `/etc/inittab` file contains the following four fields:

id:rstate:action:process

Figure 9-6 shows an `inittab` entry.

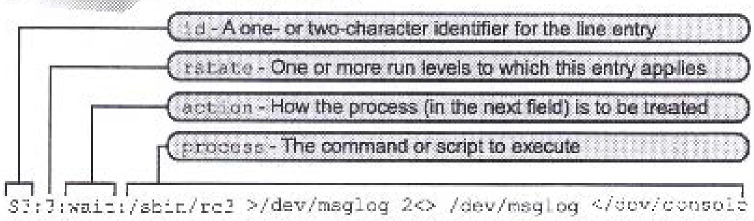

Figure 9-6 An `/etc/inittab` File Entry

Note – Message output from rc scripts is directed to the /dev/msglog file. Prior to the Solaris 8 OE, all of these messages were written to the /dev/console file. The /dev/msglog file is used for message output collection from system startup or background applications.

Table 9-2 shows an explanation for each keyword.

Table 9-2 The action Field Keywords

Keyword	Explanation
initdefault	Identifies the default run level. Read when the init process is initially invoked. Used by the init process to determine which run level to enter initially. The default is run level 3.
sysinit	Executes the process before the init process tries to access the console (for example, the console login prompt). The init process waits for completion of the process before it continues to read the inittab file.
wait	Starts a process and waits for it to complete before moving to the next entry that contains the same run level.
respawn	If the process dies, the init process restarts it. If the process does not exist, the init process starts it and continues reading the inittab file. If the process does exist, no action is required, and the init process continues reading the inittab file.
powerfail	Executes the process only if the init process receives a power fail signal.

Caution – If the rstate field is empty and the initdefault line is used, the rstate field is interpreted as 0123456, and the init process enters run level 6 as the default. This causes the system to reboot continuously.

Note – Information about additional action keywords is available in the inittab man page.

The following is an example of a default /etc/inittab file.

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
ap::sysinit:/sbin/soconfig -f /etc/sock2path
fs::sysinit:/sbin/rcS sysinit >/dev/msglog 2<>/dev/msglog </dev/console
is:3:initdefault:
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog
2<>/dev/msglog
sS:s:wait:/sbin/rcS          >/dev/msglog 2<>/dev/msglog </dev/console
s0:0:wait:/sbin/rc0          >/dev/msglog 2<>/dev/msglog </dev/console
s1:1:respawn:/sbin/rc1       >/dev/msglog 2<>/dev/msglog </dev/console
s2:23:wait:/sbin/rc2         >/dev/msglog 2<>/dev/msglog </dev/console
s3:3:wait:/sbin/rc3          >/dev/msglog 2<>/dev/msglog </dev/console
s5:5:wait:/sbin/rc5          >/dev/msglog 2<>/dev/msglog </dev/console
s6:6:wait:/sbin/rc6          >/dev/msglog 2<>/dev/msglog </dev/console
fw:0:wait:/sbin/uadmin 2 0   >/dev/msglog 2<>/dev/msglog </dev/console
of:5:wait:/sbin/uadmin 2 6   >/dev/msglog 2<>/dev/msglog </dev/console
rb:6:wait:/sbin/uadmin 2 1   >/dev/msglog 2<>/dev/msglog </dev/console
sc:234:respawn:/usr/lib/saf/sac -t 300
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: "
-T sun -d /dev/console -l console -m ldterm,ttcompat
```

The following describes each of the lines in the `inittab` file in order:

1. Initializes the streams modules
2. Configures the socket transport providers
3. Initializes the file systems
4. Defines the default run level
5. Describes a power fail shutdown
6. Defines single-user mode
7. Defines run level 0
8. Defines run level 1
9. Defines run level 2
10. Defines run level 3
11. Defines run level 5
12. Defines run level 6
13. Defines the transition to firmware
14. Defines the transition to power off
15. Defines the transition to reboot
16. Initializes the service access controller
17. Initializes the `ttymon` port monitor, which places a command-line login prompt to the console

The init Process

Figure 9-7 shows the process of bringing a system to the default run level 3.

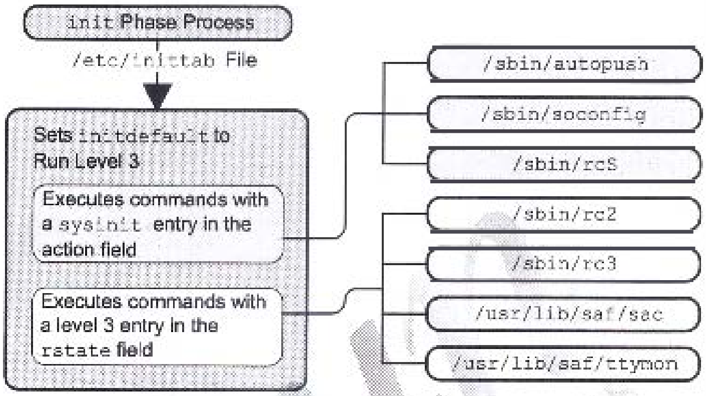

Figure 9-7 The init Process

The init process reads the /etc/inittab file to do the following:

1. Identify the initdefault entry, which defines the default run level, 3.
2. Execute any process entries that have sysinit in the action field so that any special initialization can take place before users log in. This includes the execution of /sbin/rcS, which mounts and checks the / (root), /usr, /var, and /var/adm file systems.
3. Execute any process entries that have 3 in the rstate field and an appropriate keyword in the action field, which match the default run level, 3.

   The commands executed at this run level include:

   - /usr/sbin/shutdown – The init process runs the shutdown command only if the system has received a power fail signal.
   - /sbin/rc2 – Starts the system daemons, bringing the system up into run level 2 (multiuser mode).
   - /sbin/rc3 – Starts NFS and other network resource servers for run level 3.

- `/usr/lib/saf/sac` – Starts the port monitors for devices, such as ASCII terminals and modems.
- `/usr/lib/saf/ttymon` – Starts the `ttymon` process that monitors the console for login requests. The default `terminal_type` on all systems as listed in the `/etc/inittab` file is sun.

Controlling Boot Processes

The Solaris OE provides a series of run control (rc) scripts to stop and start processes typically associated with run levels.

The /sbin Directory

Each run level has an associated rc script located in the /sbin directory. Figure 9-8 shows the rc scripts associated with each run level in the /sbin directory and their inode numbers.

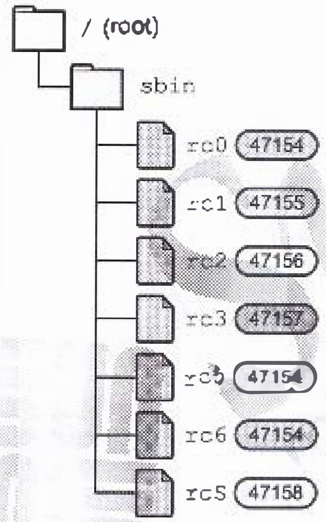

Figure 9-8 The /sbin Directory With Inode Numbers

The rc scripts are executed by the init process to set up variables, test conditions, and make calls to other scripts that start and stop processes for that run level.

The rc scripts rc0, rc5, and rc6 are hard-linked to each other. Notice each script is assigned the same inode number.

The following is an example of the hard links:

```
# cd /sbin
# ls -i rc*
47154 rc0        47156 rc2        47154 rcS        47158 rcS
47155 rc1        47157 rc3        47154 rc6
```

The Solaris OE provides the same series of rc scripts in the /etc directory for backward compatibility. These scripts are symbolic link files to the rc scripts in the /sbin directory.

The following example shows this connection:

```
# cd /etc
# ls -l rc?
lrwxrwxrwx   1 root     root          11 Feb 22 14:19 rc0 -> ../sbin/rc0
lrwxrwxrwx   1 root     root          11 Feb 22 14:19 rc1 -> ../sbin/rc1
lrwxrwxrwx   1 root     root          11 Feb 22 14:19 rc2 -> ../sbin/rc2
lrwxrwxrwx   1 root     root          11 Feb 22 14:19 rc3 -> ../sbin/rc3
lrwxrwxrwx   1 root     root          11 Feb 22 14:19 rc5 -> ../sbin/rc5
lrwxrwxrwx   1 root     root          11 Feb 22 14:19 rc6 -> ../sbin/rc6
lrwxrwxrwx   1 root     root          11 Feb 22 14:19 rcS -> ../sbin/rcS
```

Table 9-3 summarizes the tasks performed by each of the /sbin rc scripts.

Table 9-3 Run Control Scripts and Their Functions

rc Script	Function
/sbin/rc0	Runs the /etc/rc0.d/K* scripts and then the /etc/rc0.d/S* scripts to perform the following tasks: • Stops system services and daemons • Terminates all running processes • Unmounts all file systems Start scripts should only perform fast system cleanup functions.
/sbin/rc1	Runs the /etc/rc1.d/S* scripts to perform the following tasks: • Stops system services and daemons • Terminates all running user processes • Unmounts all remote file systems • Mounts all local file systems if the previous run level was S

Table 9-3 Run Control Scripts and Their Functions (Continued)

rc Script	Function
/sbin/rc2	Runs the /etc/rc2.d/K* scripts and then the /etc/rc2.d/S* scripts to perform the following tasks: • Mounts all local file systems if the previous run level was S • Removes any files and subdirectories in the /tmp directory • Configures system accounting • Configures the default router • Starts most of the system daemons
/sbin/rc3	Runs the /etc/rc3.d/K* scripts and then the /etc/rc3.d/S* scripts to perform the following tasks: • Cleans up the /etc/dfs/sharetab file • Shares all resources listed in the /etc/dfs/dfstab file • Starts the nfsd and mountd commands Note: K scripts are not normally present in the /etc/rc3.d directory, although if they were present, they would be run.
/sbin/rc5 /sbin/rc6	Runs the /etc/rc0.d/K* scripts and then the /etc/rc0.d/S* scripts to perform the following tasks: • Stops system services and daemons • Terminates all running processes • Unmounts all file systems • Starts scripts that should only perform fast system cleanup functions
/sbin/rcS	Runs the /etc/rcS.d scripts to bring up the system to run level S: • Establishes a minimal network • Mounts the /usr, /var, and /var/adm directories if they are separate file systems. • Sets the system name • Checks the / (root) and /usr file systems • Mounts pseudo file systems (/proc and /dev/fd) • Rebuilds the device entries for reconfiguration boots • Mounts other file systems that are required in single-user mode

The /etc/rc#.d Directories

The /etc/rc#.d directories contain additional scripts that start and stop system processes for that run level.

Figure 9-9 shows an example of /etc/rc#.d directories.

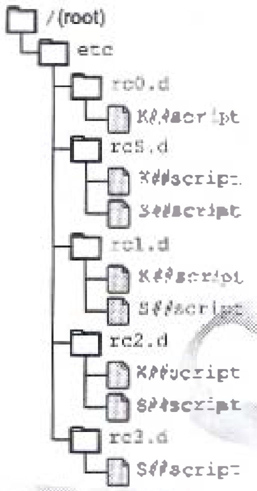

Figure 9-9 The /etc/rc#.d Directories

For example, /etc/rc2.d contains scripts to start and stop processes for run level 2. The following output shows a partial list of these scripts.

```
# ls -l /etc/rc2.d
(some output omitted)
-rwxr--r--   6 root   sys     344 Jun 19 16:56 K06mipagent
-rwxr--r--   6 root   sys     404 Jun 19 17:27 K07snmpdx
-rwxr--r--   6 root   sys    2723 Jun 19 17:00 K28nfs.server
-rwxr--r--   2 root   sys    1597 Jun 19 17:00 S20sysetup
-rwxr--r--   2 root   sys     989 Jun 19 17:12 S21perf
-rwxr-xr-x   2 root   other  1995 Jun 25 00:09 S30sysid.net
```

Start Run Control Scripts

The `/etc/rc#.d` start scripts are always run in the sort order shown by the `ls` command. The files that begin with `S` are run to start a system process. These scripts are called by the appropriate `/sbin/rc#` and this script passes the argument "`start`" to them if their names do not end in `.sh`. There are no arguments passed to `.sh` scripts. These files have names in the form of:

```
S##name-of-script
```

For example, the script that starts the line printer (LP) processes is named `S80lp`.

Stop Run Control Scripts

The `/etc/rc#.d` stop scripts (also referred to as the kill scripts) are always run in the sort order shown by the `ls` command. The files that begin with `K` are run to stop or kill a system process. These scripts are called by the appropriate `/sbin/rc#`, and this script passes the argument "`stop`" to them if their names do not end in `.sh`.

These files have names in the form of:

```
K##name-of-script
```

For example, the script that stops the NFS server processes is called `K28nfs.server`.

Note – File names that begin with a lowercase `k` or `s` are ignored by the `init` process, and they are not executed. To disable a script, rename it with the appropriate lowercase letter.

The /etc/init.d Directory

Run control scripts are located in the /etc/init.d directory.

The files shown in Figure 9-10 are hard-linked to corresponding run control scripts in the /etc/rc#.d directories.

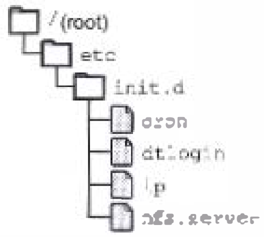

Figure 9-10 The /etc/init.d Directory

The run control script /etc/init.d/cron is hard-linked to the corresponding run control script /etc/rc2.d/S75cron, as shown by the ls commands:

```
# cd /etc/init.d
# ls -i cron
   176404 cron
# cd /etc/rc2.d
# ls -i S75cron
   176404 S75cron
```

The benefit of having individual scripts for each run level is that you can run scripts in the /etc/init.d directory individually as the root user.

You can stop a process or start a process without changing the system's run level.

For example, to stop and restart the LP print services, run the following scripts with a stop or start argument:

```
# /etc/init.d/lp stop

# /etc/init.d/lp start
```

Creating New Run Control Scripts

You can create new scripts to start and stop additional processes or services to customize a system.

For example, to eliminate the requirement for a manual start of a database server, you could create a script to start the database server automatically after the appropriate network services have started.

You could then create another script to terminate this service and shut down the database server before the network services are stopped.

To add run control scripts to start and stop a service, create the script in the /etc/init.d directory and create links in the appropriate /etc/rc#.d directory for the run level in which the service is to be started and stopped.

Refer to the README file in each /etc/rc#.d directory for more information on run control scripts.

The following procedure describes how to add a run control script:

1. Create the script in the /etc/init.d directory.

```
# vi /etc/init.d/filename
# chmod 744 /etc/init.d/filename
# chgrp sys /etc/init.d/filename
```

2. Create links to the appropriate /etc/init.d directory:

```
# cd /etc/init.d
# ln filename /etc/rc#.d/S##filename
# ln filename /etc/rc#.d/K##filename
```

3. Use the ls command to verify that the script has links in the appropriate directories.

```
# ls -li /etc/init.d/filename
# ls -li /etc/rc#.d/S##filename
# ls -li /etc/rc#.d/K##filename
```

4. Test the filename by performing the following commands:

```
# /etc/init.d/filename start
```

Figure 9-11 shows the run-level transitions that occur during the process of a system bootup or shutdown.

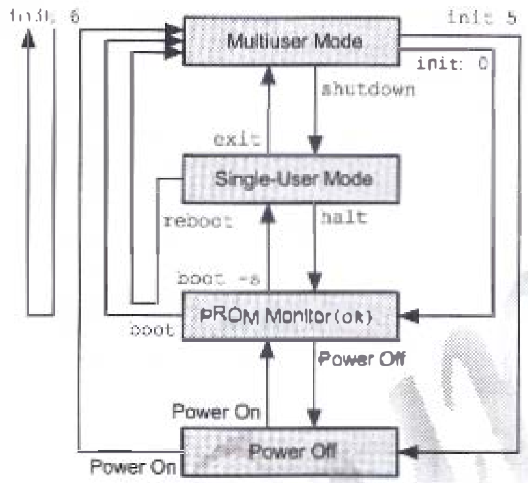

Figure 9-11 Run-Level Transitions

Note – The halt and reboot commands shown in Figure 9-11 do not process the inittab file as the init and shutdown commands do. The init and shutdown commands are the preferred methods for transitioning between run states.

Performing System Shutdown Procedures

You can shut down the Solaris OE to perform administration tasks or maintenance activities if you are anticipating a power outage or if you need to move the system to a new location.

The Solaris OE requires a clean and orderly shutdown, which stops processes, writes data in memory to disks, and unmounts file systems.

Of course, the type of work you need to do while the system is shut down determines how the system is shut down and which command you use.

The following describes the different types of system shutdowns.

- Shut down the system to single-user mode
- Shut down the system to stop the Solaris OE, and display the ok prompt
- Shut down the system and turn off power
- Shut down the system and automatically reboot to multiuser mode

The commands available to the root user for doing these types of system shutdown procedures include:

- `/sbin/init` (using run levels S, 0, 1, 5, or 6)
- `/usr/sbin/shutdown` (using run levels S, 0, 1, 5, or 6)
- `/usr/sbin/halt`
- `/usr/sbin/reboot`
- `/usr/sbin/poweroff`

Note – The `init` command accepts more arguments than those listed here. These arguments are not listed here because they are outside of the topic of system shutdown procedures.

The /usr/sbin/init Command

You use the init command to shut down, power off, or reboot a system in a clean and orderly manner. It executes the rc0 kill scripts. However, this command does not warn logged-in users that the system is being shut down, and there is no grace period.

To shut down the system to single-user mode, use either run level S or 1.

```
# init S
```

To shut down the system to stop the Solaris OE and display the ok prompt, perform the command:

```
# init 0
```

To shut down the system and turn its power off, perform the command:

```
# init 5
```

To shut down the system and then reboot to multiuser mode, perform the command:

```
# init 6
```

The /usr/sbin/shutdown Command

The shutdown command is a script that invokes the init daemon to shut down, power off, or reboot the system. It executes the rc0 kill scripts to shut down processes and applications gracefully. But unlike the init command, the shutdown command does the following:

- Notifies all logged-in users that the system is being shut down
- Delays the shutdown for 60 seconds by default
- Enables you to include an optional descriptive message to inform your users of what will transpire

The command format for the shutdown command is:

```
shutdown -y -g grace-period -i init-state
optional message
```

The -y option pre-answers the final shutdown confirmation question so that the command runs without your intervention.

The -g *grace-period* allows you to change the number of seconds from the 60-second default.

The -i *init-state* specifies the run-level that the init process is to attain. By default, system state S is used.

Note – If the shutdown command displays the error message: "shutdown: 'i' - unknown flag," it indicates that the shell has located and executed the /usr/ucb/shutdown command. Reissue the command using its full path (for example, /usr/sbin/shutdown), or set the PATH variable to ensure /usr/sbin comes before /usr/ucb.

To shut down the system to single-user mode, enter the shutdown command without options.

```
# shutdown
```

To shut down the system to stop the Solaris OE, and display the ok prompt, perform the command:

```
# shutdown -i0
```

To shut down the system and turn off its power automatically, perform the command:

```
# shutdown -i5
```

To shut down the system and then reboot to multiuser mode, perform the command:

```
# shutdown -i6
```

The -i option can be used with other command options. For example, to shut down the system and then reboot to multiuser mode, answer yes to the questions presented, provide a grace period of two minutes, and provide a message to the users, perform the command:

```
# shutdown -y -g120 -i6 "The system is being rebooted"
```

The /usr/sbin/halt Command

The halt command performs an immediate system shutdown. It does not execute the rc0 kill scripts. It does not notify logged-in users, and there is no grace period.

To shut down the system, stop the Solaris OE, and display the ok prompt, perform the command:

```
# halt
```

The /usr/sbin/poweroff Command

The poweroff command performs an immediate shutdown. It does not execute the rc0 kill scripts. It does not notify logged-in users, and there is no grace period.

To shut down the system and turn off its power, perform the command:

```
# poweroff
```

The /usr/sbin/reboot Command

The reboot command performs an immediate shutdown and reinitialization, bringing the system to run level 3 by default. The reboot command differs from the init 6 command because it does not execute the rc0 kill scripts.

To shut down the system and then reboot to multiuser mode, perform the reboot command without options:

```
# reboot
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Controlling the Boot Process (Level 1)

In this exercise, you create a new startup script, make changes in the /etc/system file, and observe their effects.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Your instructor should provide you with instructions on how to obtain a script called "banner" that will be used during this exercise.

Tasks

Complete the following tasks:

- In the /etc/rc2.d directory, create a hard link to the /etc/init.d/banner file, called S22banner. In the /etc/rcS.d directory, create a hard link to the /etc/init.d/banner file called K99banner.

  (Steps 1–5 in the Level 2 lab)

- Reboot the system, and verify that S22banner runs. Shut down the system to run level S, and verify that K99banner runs. Change back to run level 3. Make a backup copy of the /etc/system file. Check if any instances of the st driver are loaded. Modify the /etc/system file to force-load the st driver. Reboot the system, and verify that st driver instances are loaded.

  (Steps 6–10 in the Level 2 lab)

- Edit the /etc/system file to exclude the boot disk driver for your system (either dad or sd). Shut down the system to run level 0, and attempt to boot it. Make note of what happens. Interactively boot your system, and return it to an operational state.

  (Steps 11–14 in the Level 2 lab)

Exercise: Controlling the Boot Process (Level 2)

In this exercise, you create a new startup script, make changes in the /etc/system file, and observe their effects.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Your instructor should provide you with instructions on how to obtain a script called banner that will be used during this exercise.

Task Summary

In this exercise, you accomplish the following:

- In the /etc/rc2.d directory, create a hard link to the /etc/init.d/banner file, called S22banner. In the /etc/rcS.d directory, create a hard link to the /etc/init.d/banner file called K99banner.
- Reboot the system, and verify that S22banner runs. Shut down the system to run level S, and verify that K99banner runs. Change back to run level 3. Make a backup copy of the /etc/system file. Check if any instances of the st driver are loaded. Modify the /etc/system file to force-load the st driver. Reboot the system, and verify that st driver instances are loaded.
- Edit the /etc/system file to exclude the boot disk driver for your system (either dad or sd). Shut down the system to run level 0, and attempt to boot it. Make note of what happens. Boot the system using the -a option of the boot command. Use your backup of the /etc/system file as required. Replace the /etc/system file with your backup when finished, and reboot the system.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to /etc/init.d. Make sure that the banner script your instructor provided you is present and executable.
2. Verify that the script runs with both the start and stop arguments.
3. Change the directory to /etc/rc2.d. Create a hard link called S22banner that points to the same data as the /etc/init.d/banner file.
4. Change the directory to the /etc/rcS.d directory. Create a hard link called K99banner that points to the same data as the /etc/init.d/banner file.
5. Reboot the system, and watch for the output of the script you just installed.

   Does the startup message from S22banner appear?
6. Log in as the root user, and open a terminal window. Use the init command to change to run level S.

   Does the shutdown message from K99banner appear?
7. Type the password for the root user to log in at the command line. Change to run level 3.
8. Log in as the root user, and open a terminal window. Change the directory to /etc.
9. Make a backup copy of the /etc/system file, and name the backup file system.orig.
10. If your system uses a SCSI tape device, perform the following:
    a. Log in as the root user, and open a terminal window. Use the prtconf command to list instances of the st driver currently loaded.

       How many instances are reported?
    b. Edit the /etc/system file so that it includes the following line:

       forceload: drv/st

       Then reboot the system.
    c. Log in as root, and open a terminal window. Again list instances of the st driver currently loaded.

       How many instances are reported?

11. Edit the /etc/system file so that it excludes the main disk driver for your system.

    On systems using SCSI disks, add the following:

    exclude: drv/sd

    On systems using IDE disks, add the following:

    exclude: drv/dad

12. Shut down the system to run level 0, and then attempt to boot it again.

    What happened?

13. Use the boot -a command to boot the system, and supply the name of your backup file called etc/system.orig (note there is *not* a leading slash to the etc). Press the Return key to accept the default values for all other boot parameters. For example:

```
ok boot -a
Enter filename [kernel/sparcv9/unix]: <Return>
Enter default directory for modules [/platform...]: <Return>
Name of system file [etc/system]: etc/system.orig
root filesystem type [ufs]: <Return>
Enter physical name of root device [/...]: <Return>
```

14. Log in as the root user, and open a terminal window. Copy the /etc/system.orig file to the /etc/system file. Reboot the system.

Exercise: Controlling the Boot Process (Level 3)

In this exercise, you create a new startup script, make changes in the /etc/system file, and observe their effects.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Your instructor should provide you with instructions on how to obtain a script called banner that will be used during this exercise.

Task Summary

In this exercise, you accomplish the following:

- In the /etc/rc2.d directory, create a hard link to the /etc/init.d/banner file, called S22banner. In the /etc/rcS.d directory, create a hard link to the /etc/init.d/banner file called K99banner.
- Reboot the system, and verify that S22banner runs. Shut down the system to run level S, and verify that K99banner runs. Change back to run level 3. Make a backup copy of the /etc/system file. Check if any instances of the st driver are loaded. Modify the /etc/system file to force-load the st driver. Reboot the system, and verify that st driver instances are loaded.
- Edit the /etc/system file to exclude the boot disk driver for your system (either dad or sd). Shut down the system to run level 0, and attempt to boot it. Make note of what happens. Boot the system using the -a option of the boot command. Use your backup of the /etc/system file as required. Replace the /etc/system file with your backup when finished, and reboot the system.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to /etc/init.d. Make sure that the banner script your instructor provided you is present and executable.

```
# cd /etc/init.d
# ls -l banner
# chmod 744 banner
```

2. Make the banner script executable, and verify that it runs with both the start and stop arguments.

```
# ./banner start
# ./banner stop
```

3. Change the directory to the /etc/rc2.d directory. Create a hard link called S22banner that points to the same data as the /etc/init.d/banner file.

```
# cd /etc/rc2.d
# ln /etc/init.d/banner S22banner
```

4. Change the directory to the /etc/rcS.d directory. Create a hard link called K99banner that points to the same data as the /etc/init.d/banner file.

```
# cd /etc/rcS.d
# ln /etc/init.d/banner K99banner
```

5. Reboot the system, and watch for the output of the script you just installed.

```
# init 6
```

Does the startup message from S22banner appear?

Yes.

6. Log in as the root user, and open a terminal window. Use the init command to change to run level S.

```
# init S
```

Does the shutdown message from K99banner appear?

Yes.

7. Type the password for the root user to log in at the command line. Change to run level 3.

```
# init 3
```

8. Log in as the root user, and open a terminal window. Change the directory to /etc.

```
# cd /etc
```

9. Make a backup copy of the /etc/system file, and name the backup file system.orig.

```
# cp system system.orig
```

10. If your system uses a SCSI tape device, perform the following:

    a. Log in as the root user, and open a terminal window. Use the prtconf command to list instances of the st driver currently loaded.

```
# prtconf | grep "st, instance"
```

    How many instances are reported?

    None

    b. Edit the /etc/system file so that it includes the following line:

    forceload: drv/st

    Then reboot the system.

```
# init 6
```

    c. Log in as root, and open a terminal window. Again list instances of the st driver currently loaded.

```
# prtconf | grep "st, instance"
```

    How many instances are reported?

    The number varies depending on how many SCSI controllers are present. You should see instances 0 through 6 for a system with one controller.

11. Edit the /etc/system file so that it excludes the main disk driver for your system.

    On systems using SCSI disks, add the following:

    exclude: drv/sd

    On systems using IDE disks, add the following:

    exclude: drv/dad

12. Shut down the system to run level 0, and then attempt to boot it again.

```
# shutdown -y -i0 -g0
(shutdown messages)
ok boot
```

What happened?

The system is unable to boot. Excluding this driver prevents you from using the boot disk so long as you use the same /etc/system file. You must boot using the -a option to be able to supply an alternative file for the /etc/system file.

13. Use the boot -a command to boot the system, and supply the name of your backup file called etc/system.orig (Note there is not a leading slash to the etc). Press Return to accept the default values for all other boot parameters. For example:

```
ok boot -a
Enter filename [kernel/sparcv9/unix]: <Return>
Enter default directory for modules [/platform...]: <Return>
Name of system file [etc/system]: etc/system.orig
root filesystem type [ufs]: <Return>
Enter physical name of root device [/...]: <Return>
```

14. Log in as the root user and open a terminal window. Copy the /etc/system.orig file to the /etc/system file. Reboot the system.

```
# cd /etc
# cp system.orig system
# init 6
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 10

Performing User Administration

Objectives

Upon completion of this module, you should be able to:

- Describe user administration fundamentals
- Manage user accounts
- Manage initialization files

The following course map shows how this module fits into the current instructional goal.

Performing User and Security Administration

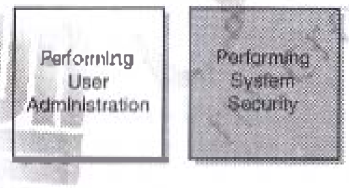

Figure 10-1 Course Map

Introducing User Administration

An important system administration task is setting up user accounts for each user who requires system access. Each user needs a unique account name, a user identification (UID) number, a home directory, and a login shell. You also have to determine which groups a user may access.

Main Components of a User Account

The following is a list of the main components of a user account:

- User name – A unique name that a user enters to log in to a system. The user name is also called the login name.
- Password – A combination of six to eight letters, numbers, or special characters that a user enters with the login name to gain access to a system.
- UID number – A user account's unique numerical identification within the system.
- Group identification (GID) number – A unique numerical identification of the group to which the user belongs.

Note – You can add a user to predefined groups listed in the `/etc/group` file.

- Comment – Information that identifies the user. A comment generally contains the full name of the user and optional information, such as a phone number or a location.
- User's home directory – A directory into which the user is placed after login. The directory is provided to the user to store and create files.
- User's login shell – The user's work environment is set up by the initialization files that are defined by the user's login shell.
- Password aging – An optional feature to require users to change their passwords on a regular basis.

System Files That Store User Account Information

The Solaris™ Operating Environment (Solaris OE) stores user account and group entry information in the following system files:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`

Authorized system users have login account entries in the `/etc/passwd` file.

The `/etc/shadow` file is a separate file that contains the encrypted passwords. To further control user passwords, you can enforce password aging. This information is also maintained in the `/etc/shadow` file.

The `/etc/group` file defines the default system group entries. You use this file to create new group entries or modify existing group entries on the system.

The `/etc/passwd` File

Due to the critical nature of the `/etc/passwd` file, you should refrain from editing this file directly. Instead, you should use the Solaris™ Management Console or command-line tools to maintain the file.

The following is an example of an `/etc/passwd` file that contains the default system account entries.

```
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
```

Each entry in the /etc/passwd file contains seven fields. A colon separates each field. The following is the format for an entry:

```
loginID:x:UID:GID:comment:home_directory:login_shell
```

Table 10-1 defines the requirements for each of the seven fields.

Table 10-1 Fields in the /etc/passwd File

Field	Description
loginID	Represents the user's login name. It should be unique to each user. The field should contain a string of no more than eight letters (A–Z, a–z) and numbers (0–9). The first character should be a letter, and at least one character should be lowercase. **Note** – Even though some programs allow a maximum of 32 characters, as well as user names that contain periods (.), underscores (_), and hyphens (-), this practice is not recommended and might cause problems with other programs.
x	Represents a placeholder for the user's encrypted password, which is kept in the /etc/shadow file.
UID	Contains the UID number used by the system to identify the user. UID numbers for users range from 100 to 60000. Values 0 through 99 are reserved for system accounts. UID number 60001 is reserved for the nobody account. UID number 60002 is reserved for the noaccess account. While duplicate UID numbers are allowed, they should be avoided unless absolutely required by a program. **Note** – The maximum value for a UID is 2147483647. However, the UIDs over 60000 do not have full utility and are incompatible with some Solaris OE features. Avoid using UIDs over 60000 so as to be compatible with earlier versions of the operating environment.

Table 10-1 Fields in the /etc/passwd File (Continued)

Field	Description
GID	Contains the GID number used by the system to identify the user's primary group. GID numbers for users range from 100 to 60000. (Those between 0 and 99 are reserved for system accounts.)
comment	Typically contains the user's full name.
home_directory	Contains the full path name to the user's home directory.
login_shell	Defines the user's login shell. There are six possible login shells in the Solaris OE: the Bourne shell, the Korn shell, the C shell, the Z shell, the BASH shell, and the TC shell.

Table 10-2 shows the default system account data for entries in the /etc/passwd file.

Table 10-2 Default System Account Entries

User Name	User ID	Description
root	0	The root account that has access to the entire system. It has almost no restrictions and overrides all other logins, protections, and permissions.
daemon	1	The system daemon account that is associated with routine system tasks.
bin	2	The administrative daemon account that is associated with running system binary files.
sys	3	The administrative daemon account that is associated with system logging or updating files in temporary directories.
adm	4	The administrative daemon account that is associated with system logging.
lp	71	The line printer (lp) daemon account.
uucp	5	The daemon account associated with UNIX®-to-UNIX Copy Protocol (UUCP) functions.
nuucp	6	The account that is used by remote systems to log in to the host and start file transfers.

Table 10-2 Default System Account Entries (Continued)

User Name	User ID	Description
smmsp	25	The sendmail message submission daemon account.
listen	37	The network listener daemon account.
nobody	60001	The anonymous user account that is assigned by a Network File System (NFS) server when an unauthorized root user makes a request. The nobody user account is assigned to software processes that do not need any special permissions.
noaccess	60002	The account assigned to a user or a process that needs access to a system through some application instead of through a system login procedure.
nobody4	65534	The anonymous user account that is the SunOS™ 4.0 or 4.1 software version of the nobody account

Note – The nobody account secures NFS resources. When a user is logged in as root on an NFS client and attempts to access a remote file resource, the UID number changes from 0 to the UID of nobody (60001).

The /etc/shadow File

Due to the critical nature of the /etc/shadow file, you should refrain from editing it directly. Instead, maintain the fields of the file by using the Solaris Management Console or command-line tools. Only the root user can read the /etc/shadow file.

The following is an example /etc/shadow file that contains initial system account entries.

```
root:5RiJS.yvdGBkU:6445::::::
daemon:NP:6445::::::
bin:NP:6445::::::
sys:NP:6445::::::
adm:NP:6445::::::
lp:NP:6445::::::
uucp:NP:6445::::::
nuucp:NP:6445::::::
smmsp:NP:6445::::::
```

```
listen:*LK*:::::::
nobody:NP:6445::::::
noaccess:NP:6445::::::
nobody4:NP:6445::::::
```

Each entry in the `/etc/shadow` file contains nine fields. A colon separates each field. The ninth field is reserved for future use and is not currently used.

Following is the format of an entry:

loginID:password:lastchg:min:max:warn:inactive:expire:

Table 10-3 defines the requirements for each of the eight fields.

Table 10-3 Fields in the `/etc/shadow` File

Field	Description
loginID	The user's login name.
password	A 13-character encrypted password. The string `*LK*` indicates a locked account, and the string `NP` indicates no valid password. Passwords must be constructed to meet the following requirements: Each password must be at least six characters and contain at least two alphabetic characters and at least one numeric or special character. It cannot be the same as the login ID or the reverse of the login ID.
lastchg	The number of days between January 1, 1970, and the last password modification date.
min	The minimum number of days required between password changes.
max	The maximum number of days the password is valid before the user is prompted to enter a new password at login.
warn	The number of days the user is warned before the password expires.
inactive	The number of inactive days allowed for the user before the user's account is locked.

Table 10-3 Fields in the /etc/shadow File (Continued)

Field	Description
expire	The date (given as number of days since January 1, 1970) when the user account expires. After the date is exceeded, the user can no longer log in.

The /etc/group File

Each user belongs to a group that is referred to as the user's primary group. The GID number, located in the user's account entry within the /etc/passwd file, specifies the user's primary group.

Each user can also belong to up to 15 additional groups, known as secondary groups. In the /etc/group file, you can add users to group entries, thus establishing the user's secondary group affiliations.

The following is an example of the default entries in an /etc/group file:

```
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
sysadmin::14:
smmsp::25:smmsp
nobody::60001:
noaccess::60002:
nogroup::65534:
```

Each line entry in the /etc/group file contains four fields. A colon character separates each field. The following is the format for an entry:

```
groupname:group-password:GID:username-list
```

Table 10-4 defines the requirements for each of the four fields.

Table 10-4 Fields in the /etc/group File

Field	Description
groupname	Contains the name assigned to the group. Group names contain up to a maximum of eight characters.
group-password	Usually contains an empty field or an asterisk. This is a relic of earlier versions of UNIX. A group-password is a security hole because it might allow an unauthorized user who is not a member of the group but who knows the group password, to enter the group. **Note –** The newgrp command changes a user's primary group association within the shell environment from which it is executed. If this new, active group has a password and the user is not a listed member in that group, the user must enter the password before the newgrp command can continue.
GID	Contains the group's GID number. It is unique on the local system and should be unique across the organization. Numbers 0 to 99, 60001, 60002 and 65534 are reserved for system group entries. User-defined groups range from 100 to 60000.
username-list	Contains a comma-separated list of user names that represent the user's secondary group memberships. By default, each user can belong to a maximum of 15 secondary groups. **Note –** The maximum number of groups is set by the kernel parameter called ngroups_max. You can set this parameter in the /etc/system file to allow for a maximum of 32 groups. Not all applications will be able to reference group memberships greater than 16. NFS is a notable example.

The `/etc/default/passwd` File

Set values for the following parameters in the `/etc/default/passwd` file to control properties for all users' passwords on the system:

- `MAXWEEKS` – Sets the maximum time period (in weeks) that the password is valid.
- `MINWEEKS` – Sets the minimum time period before the password can be changed.
- `PASSLENGTH` – Sets the minimum number of characters for a password. Valid entries are 6, 7, and 8.
- `WARNWEEKS` – Sets the time period prior to a password's expiration to warn the user that the password will expire.

Note – The `WARNWEEKS` value does not exist by default in the `/etc/default/passwd` file, but it can be added.

The password aging parameters `MAXWEEKS`, `MINWEEKS`, and `WARNWEEKS` are default values. If set in the `/etc/shadow` file, the parameters in that file override those in the `/etc/default/passwd` file for individual users.

Managing User Accounts

Each of the following sections present two sets of command-line tools for managing user accounts: the command-line tools used in the Solaris OE versions prior to the Solaris 9 OE, and the new set of command-line tools developed for the Solaris 9 OE.

Introducing Command-Line Tools

The Solaris 7 OE and the Solaris 8 OE provide you with command-line tools, defined as follows:

- useradd – Adds a new user account on the local system
- usermod – Modifies a user's account on the local system
- userdel – Deletes a user's account from the local system
- groupadd – Adds a new group entry to the system
- groupmod – Modifies a group entry on the system
- groupdel – Deletes a group entry from the system

In addition to these command-line tools, the Solaris 9 OE has a new set of command-line tools that accomplish the same tasks. They are the smuser and smgroup commands.

The smuser command enables you to manage one or more users on the system with the following set of subcommands:

- add – Adds a new user account
- modify – Modifies a user's account
- delete – Deletes a user's account
- list – Lists one or more user entries

Note – The `smuser` and `smgroup` commands are the command-line interface equivalent to the Solaris Management Console range of operation, and allow you to perform Solaris Management Console actions in scripts. Therefore, the `smuser` and `smgroup` commands have numerous subcommands and options designed to function across domains and multiple systems. This module describes only the basic commands.

The `smgroup` command enables you to manage one or more groups on the system with the following set of subcommands:

- `add` – Adds a new group entry
- `modify` – Modifies a group entry
- `delete` – Deletes a group entry
- `list` – Lists one or more group entries

Any subcommand to add, modify, list, or delete users with the `smuser` and `smgroup` commands requires authentication with the Solaris Management Console server and requires the initialization of the Solaris Management Console. For example, the following is the command format for the `smuser` command:

```
/usr/sadm/bin/smuser subcommand [auth_args] -- [subcommand_args]
```

The authorization arguments are all optional. However, if you do not specify the authorization argument, the system might prompt you for additional information, such as a password for authentication purposes.

The `--` option separates the subcommand-specific options from the authorization arguments. The `--` option must be entered even if an authorization argument is not specified because it must precede the subcommand arguments.

The subcommand arguments are quite numerous. For a complete listing of the subcommands, refer to the `smuser` man page. It is important to note that descriptions and other arguments that contain white space must be enclosed in double quotation marks.

Creating a User Account

Use the useradd or smuser add command to add new user accounts to the local system. These commands add an entry for a new user into the /etc/passwd and /etc/shadow files.

These commands also automatically copy all the initialization files from the /etc/skel directory to the user's new home directory.

The useradd Command Format and Options

The following is the command format for the useradd command:

```
useradd [ -u uid ][ -g gid ][ -G gid [,gid,.. ]]
[ -d dir ][ -m ][ -s shell ][ -c comment ] loginname
```

Table 10-5 shows the options for the useradd command.

Table 10-5 Options for the useradd Command

Option	Definition
-u uid	Sets the UID number for the new user
-g gid	Defines the new user's primary group
-G gid	Defines the new user's secondary group memberships
-d dir	Defines the full path name for the user's home directory
-m	Creates the user's home directory if it does not already exist
-s shell	Defines the full path name for the shell program of the user's login shell
-c comment	Specifies any comment, such as the user's full name and location
loginname	Defines the user's login name for the user account
-D	Displays the defaults that are applied to the useradd command

The following example uses the useradd command to create an account for a user named newuser1. It assigns 100 as the UID number, adds the user to the group other, creates a home directory in the /export/home directory, and sets /bin/ksh as the login shell for the user account.

```
# useradd -u 100 -g other -d /export/home/newuser1 -m -s /bin/ksh -c
"Regular User Account" newuser1
64 blocks
#
```

User accounts are locked by default when added with the useradd command.

By convention, a user's login name is also the user's home directory name.

You use the passwd command to create a password for the new account.

```
# passwd newuser1
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for newuser1
```

The smuser add Command Format and Options

The following is the command format for the smuser add command:

```
smuser add [auth_args] -- [subcommand_args]
```

Table 10-6 shows some of the most common subcommand arguments for the smuser add command.

Table 10-6 Subcommand Arguments for the smuser add Command

Subcommand Argument	Definition
-c *comment*	A short description of the login, typically the user's name. This string can be up to 256 characters.
-d *directory*	Specifies the home directory of the new user and is limited to 1024 characters.
-g *group*	Specifies the new user's primary group membership.

Table 10-6 Subcommand Arguments for the smuser add Command (Continued)

Subcommand Argument	Definition
-G *group*	Specifies the user's secondary group membership.
-n *login*	Specifies the user's login name.
-s *shell*	Specifies the full path name of the user's login shell.
-u *uid*	Specifies the user ID of the user you want to add. If you do not specify this option, the system assigns the next available unique UID greater than 100.

The following example uses the smuser add command to create an account for a user named newuser2. It designates the login name as newuser2, assigns the UID number 500, adds the user to the group other, creates a home directory in the /export/home directory, and sets /bin/ksh as the login shell for the user account.

Note – The -x autohome=N option to the smuser command adds the user without automounting the user's home directory. See the man page for automount for more information.

```
# /usr/sadm/bin/smuser add -- -n newuser2 -u 500 -g other -d
/export/home/newuser2 -c "Regular User Account 2" -s /bin/ksh -x
autohome=N
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: EnterPassword
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys41
Login to sys41 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys41 was
successful.
```

Users are added without a password by default with the smuser command. Use the passwd command to create one.

```
# passwd newuser2
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for newuser2
```

Modifying a User Account

Use the usermod or smuser modify command to modify a user's login account on the system.

The usermod Command Format and Options

The following is the command format for the usermod command:

```
usermod [ -u uid [ -o ] ] [ -g gid ] [ -G gid [ , gid . . . ] ]
[ -d dir ] [ -m ] [ -s shell ] [ -c comment ]
[ -l newlogname] loginname
```

In general, the options for the usermod command function the same as those for the useradd command.

Table 10-7 shows the key options to the usermod command.

Table 10-7 Key Options for the usermod Command

Option	Definition
-o	Allows a UID to be duplicated.
-m	Moves the user's home directory to the new location specified with the -d option.
-l *newlogname*	Changes a user's login name for the specified user account.
-f *inactive*	Sets the number of inactive days that are allowed on a user account. If the account is not logged in to for the specified number of days, it is locked.
-e *expire*	Sets an expiration date on the user account. Specifies the date (*mm/dd/yy*) on which a user can no longer log in and access the account. After that date, the account is locked.
loginname	Identifies the user's login name for the current user account.

The following example changes the login name and home directory for newuser1 to usera.

```
# usermod -m -d /export/home/usera -l usera newuser1
```

The smuser modify Command Format and Options

The following is the command format for the smuser modify command:

```
smuser modify [auth_args] -- [subcommand_args]
```

In general, the options for the smuser modify command function the same as for the smuser add command. Refer to the smuser(1M) man page for additional options.

Table 10-8 shows the options for the smuser modify command.

Table 10-8 Options for the smuser modify Command

Option	Definition
-n login	Specifies the user's login name
-N login	Specifies the user's new login name

The following example changes the login name and home directory for newuser2 to userb.

```
# /usr/sadm/bin/smuser modify -- -n newuser2 -N userb -d
/export/home/userb
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: EnterPassword
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys41
Login to sys41 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys41 was
successful.
```

Deleting a User Account

Use the `userdel` command or `smuser delete` command to delete a user's login account from the system.

The following is the command format for the `userdel` command:

```
userdel -r login
```

The `userdel` command also removes the user's home directory and all of its contents if you request it to do so. Use the `-r` option to remove the user's home directory from the local file system. This directory must exist.

The following example removes the login account for a user named `usera`.

```
# userdel usera
```

To request that both the user's account and home directory be removed from the system at the same time, perform the command:

```
# userdel -r usera
```

The `smuser delete` Command Format and Options

The following is the command format for the `smuser delete` command:

```
smuser delete [auth_args] -- [subcommand_args]
```

The following example removes the `userb` account from the system:

```
# /usr/sadm/bin/smuser delete -- -n userb
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: EnterPassword
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from sys41
Login to sys41 as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from sys41 was
successful.
```

Note – Unlike the `userdel` command, the `smuser delete` command has no `-r` equivalent option for deleting the home directory. The user's home directory must be deleted manually.

Creating a Group Entry

As the root user, you create new group entries on the local system by using the groupadd or smgroup add command. These commands add an entry for the new group into the /etc/group file. Like the smuser command, the smgroup add command uses the same subcommands and authentication arguments derived from the Solaris Management Console.

The groupadd Command Format and Options

The following is the command format for the groupadd command:

```
groupadd [ -g gid [ -o ] ] groupname
```

Table 10-9 shows the options for the groupadd command.

Table 10-9 Options for the groupadd Command

Option	Description
-g *gid*	Assigns the GID number for the new group
-o	Allows the GID number to be duplicated

The following example uses the groupadd command to create the new group class on the local system:

```
# groupadd -g 301 class
```

The smgroup add Command Format and Options

The following is the command format for the smgroup add command:

```
/usr/sadm/bin/smgroup subcommand [auth_args] -- [subcommand_args]
```

Table 10-10 shows the options for the smgroup add command.

Table 10-10 Options for the smgroup add Command

Option	Description
-g *gid*	Specifies the GID number for the new group
-m *group_member*	Specifies the new members to add to the group
-n *group_name*	Specifies the name of the new group

The following example uses the smgroup add command to create a new group called workgroup with a GID of 123, and to add usera to the group:

```
# /usr/sadm/bin/smgroup add -- -n workgroup -g 123 -m usera
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: EnterPassword
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys41
Login to sys41 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys41
was successful.
```

Modifying a Group Entry

You can use the following commands to modify a group entry:

- The groupmod command
- The smgroup modify command

The groupmod Command Format and Options

The following is the command format for the groupmod command:

```
groupmod [ -g gid [ -o ] ] [ -n name ] groupname
```

Table 10-11 defines the options for the groupmod command.

Table 10-11 Options for the groupmod Command

Options	Description
-g *gid*	Specifies the new GID number for the group
-o	Allows the GID number to be duplicated
-n *name*	Specifies the new name for the group

The following example changes the class account group GID number to 400:

```
# groupmod -g 400 class
```

The smgroup modify Command Format and Options

The following is the command format for the smgroup modify command:

```
/usr/sadm/bin/smgroup subcommand [auth_args] -- [subcommand_args]
```

Table 10-12 shows the options for the smgroup modify command.

Table 10-12 Options for the smgroup modify Command

Option	Description
-n *name*	Specifies the name of the group you want to modify
-m *new_member*	Specifies the new members to add to the group
-N *new_group*	Specifies the new group name

The following example changes the group workgroup to schoolgroup:

```
# /usr/sadm/bin/smgroup modify -- -n workgroup -N schoolgroup
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: EnterPassword
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys41
Login to sys41 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys41
was successful.
```

Deleting a Group Entry

Use the groupdel or smgroup delete commands to delete a group entry from the /etc/group file on the system.

The groupdel Command Format

The following is the command format for the groupdel command:

```
groupdel groupname
```

The following example removes the group entry class from the local system:

```
# groupdel class
```

The `smgroup delete` Command Format and Options

The following is the command format for the `smgroup delete` command:

```
/usr/sadm/bin/smgroup  subcommand [auth_args] -- [subcommand_args]
```

You can use the `-n group_name` option with the `smgroup delete` command to specify the name of the group you want to delete.

The following example deletes the group entry `schoolgroup` from the local system:

```
# /usr/sadm/bin/smgroup delete -- -n schoolgroup
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys41
Login to sys41 as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from sys41
was successful.
```

Using the Solaris Management Console Users Tool

The Solaris Management Console Users Tool is a graphical user interface (GUI) that provides access to Solaris OE system administration tools. You can use it for adding, removing, and modifying user and group entries. The following sections contain a demonstration.

Start the Solaris Management Console by typing smc on the command line or by clicking the SMC icon under the Tools submenu. After the "Welcome to Solaris Management Console" message appears, click This Computer to open the Solaris Management Console window. See Figure 10-2.

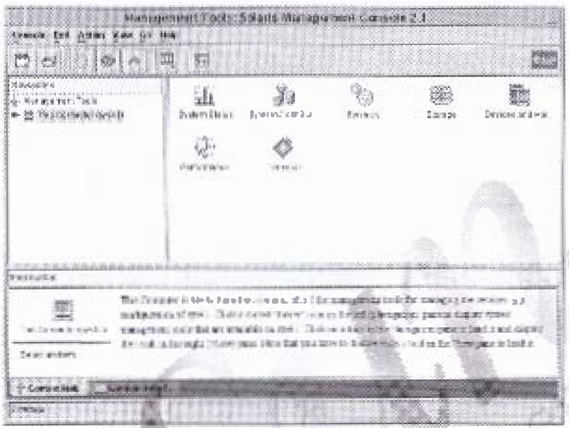

Figure 10-2 Solaris Management Console Window

Adding a User Account

The default method of adding a user account through Solaris Management Console is to add the user account with the user's home directory automounted. The following steps demonstrate how to build a user template that adds the user account with the user's directory under the /export/home directory.

To add a user account, perform the following steps:

1. Click This Computer in the Navigation pane to display the system management tools.
2. Click System Configuration to display the tool for setting up a new user account.
3. Click Users and enter the user name and password to be used for authentication if prompted to do so by Solaris Management Console.
4. Double-click User Templates to access the tool to create and manage user templates.
5. From the Menu Bar, select Add User Template from the Action list.

Figure 10-3 shows the Add User Template window.

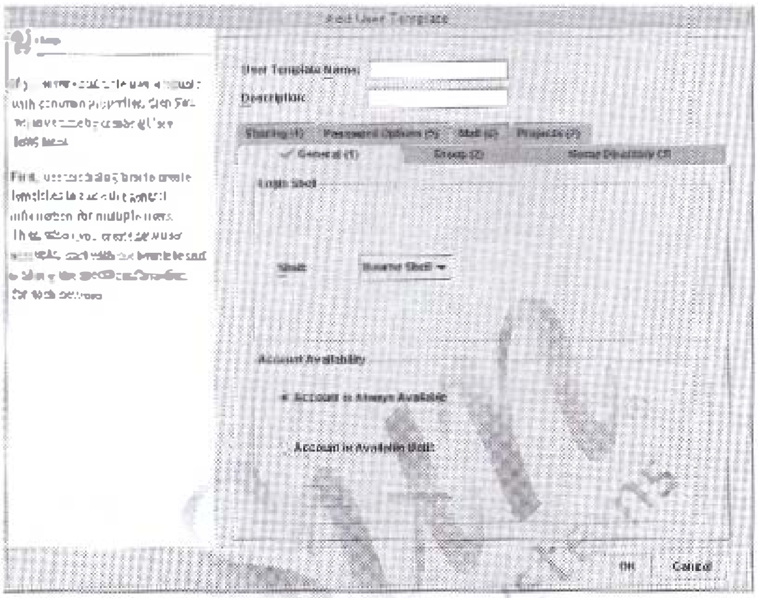

Figure 10-3 Add User Template Window

6. Type the name SA239user in the User Template Name field. You can provide an optional description if you wish.
7. Click the Home Directory tab. Type your system name in the Home Directory Server field. Uncheck the check box labeled Automatically Mount Home Directory.

Figure 10-4 shows the Add User Template window with the Home Directory Information completed.

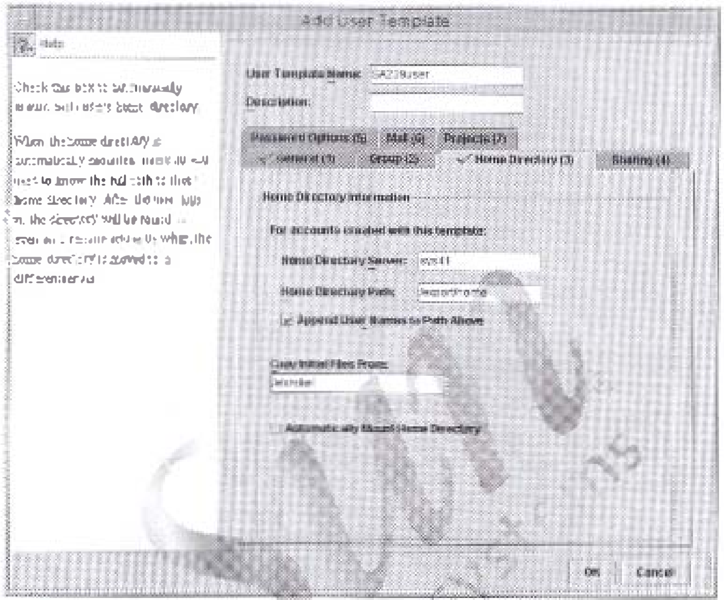

Figure 10-4 Add User Template Window (Home Directory Tab)

8. Click OK, and the Solaris Management Console (User Templates) window (Figure 10-5) reappears with the sa239user template in the View pane.

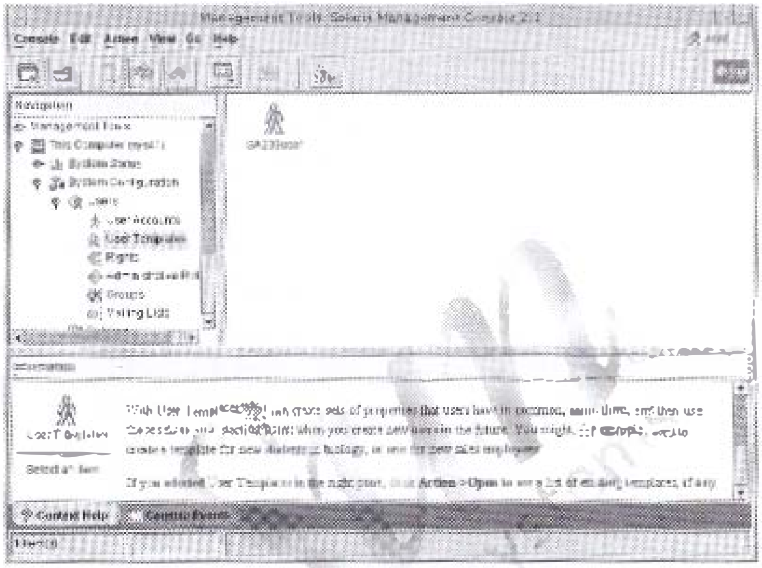

Figure 10-5 Management Tools: Solaris Management Console Window – User Templates

9. Click User Accounts from the Navigation pane, and a list of user accounts on the system appears in the View pane. See Figure 10-6.

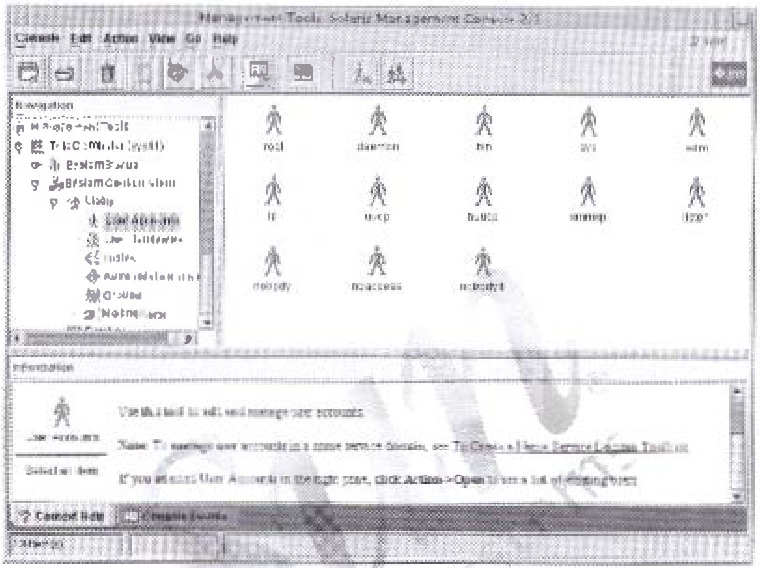

Figure 10-6 Management Tools: Solaris Management Console Window – User Accounts

10. From the Menu Bar, select Action. Then select Add User, and then select From Template. The Add User From Template window appears. See Figure 10-7.

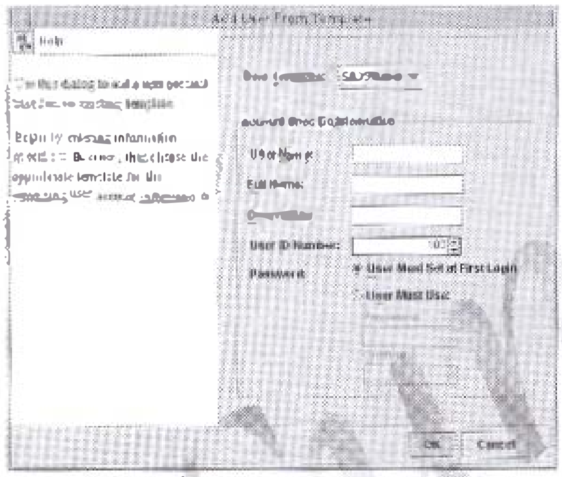

Figure 10-7 Add User From Template Window

Because you only have one template created, it is the default template available from the User Template pull-down list.

11. In the field beside User Name, enter the login ID of the user you wish to create. A full name and description are optional.
12. Click the button User Must Use and fill in the password and confirmation fields with the password 123pass.
13. Click OK and the Solaris Management Console (User Accounts) window reappears with the user account you just created in the View pane.

14. Double-click the user account you just created. The User Properties window appears (Figure 10-8). You can view and modify the properties of that user account.

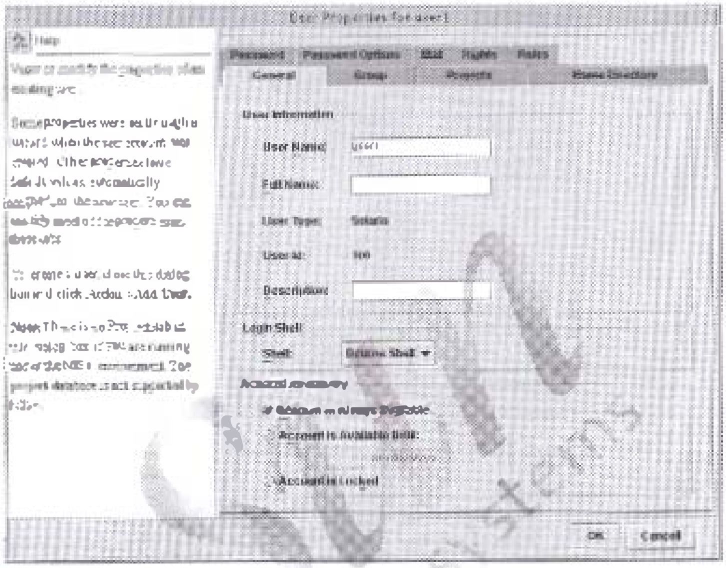

Figure 10-8 User Properties Window

15. Click the Group tab.

The screen changes to reveal a list of groups. Figure 10-9 shows the information under the Group tab, including the primary group to which the user belongs and a list of available groups.

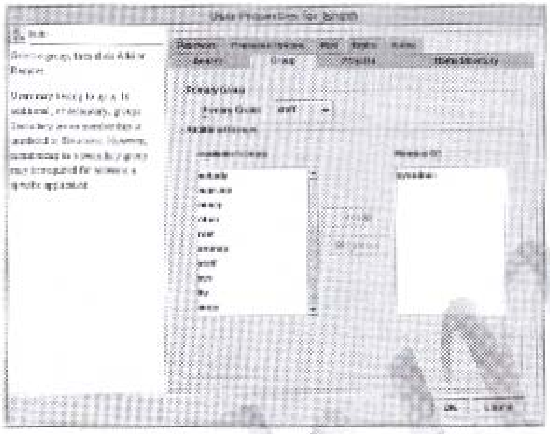

Figure 10-9 User Properties Window – Adding Groups

16. You can click a group listed under Available Groups, and then click Add, and the group moves into the Member Of column.
17. Add the groups to which you want the user to belong, and then click OK.

Deleting a User Account

Figure 10-10 shows the initial steps you take to remove a user account from the system.

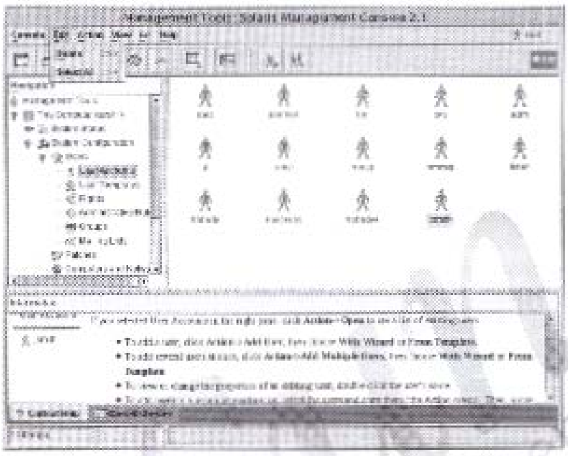

Figure 10-10 Management Tools: Solaris Management Console Window – Deleting a User Account Window

1. Highlight the user account in the User Accounts window.
2. From the Menu Bar, click Edit. Select **Delete** from the Edit menu.

   Figure 10-11 shows the warning window that appears asking you to verify that you want to delete the user account.

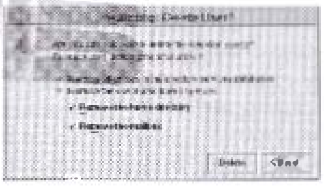

Figure 10-11 Warning: Delete User

   This window also contains options to remove the user's home directory and to remove the user's mailbox.

3. Check the appropriate boxes, and then click **Delete**. The user account is deleted.

Troubleshooting Login Issues

Some of the most common problems you might encounter as a system administrator are user login problems. There are two categories of login problems: login problems when the user logs in at the command line and login problems when the user logs in from the Common Desktop Environment (CDE).

The CDE uses more configuration files, so there are more potential problems associated with logging in from the CDE. When you troubleshoot a login problem, first determine whether you can log in from the command line. Attempt to log in from another system by using either the `telnet` command or the `rlogin` command, or click Options from the CDE login panel and select Command Line Login. If you can log in successfully at the command line, then the problem is with the CDE configuration files. If you cannot log in at the command line, then the problem is more serious and involves key configuration files.

Login Problems at the Command Line

Table 10-13 presents an overview of common login problems that occur when the user logs in at the command line.

Table 10-13 Login Problems at the Command Line

Login Problem	Description
`Login incorrect`	This message occurs when there are problems with the login information. The most common cause of an incorrect login message is a mistyped password. Make sure that the correct password is being used, and then attempt to enter it again. Remember that passwords are case-sensitive, so you cannot interchange uppercase letters and lowercase letters. In the same way, the letter "o" is not interchangeable with the numeral "0" nor is the letter "l" interchangeable with the numeral "1."
`Permission denied`	This message occurs when there are login, password, or NIS+ security problems. Most often, an administrator has locked the user's password or the user's account has been terminated.

Table 10-13 Login Problems at the Command Line (Continued)

Login Problem	Description
`Password will not work at lockscreen`	A common error is to have the Caps Lock key on, which causes all letters to be uppercase. This does not work if the password contains lowercase letters.
`No shell`	This message occurs when the user's shell does not exist, is typed incorrectly, or is wrong in the `/etc/passwd` file.
`No directory! Logging in with home=/`	This message occurs when the user cannot access the home directory for one of the following reasons: An entry in the `/etc/passwd` file is incorrect, or the home directory has been removed or is missing, or the home directory exists on a mount point that is currently unavailable.
`Choose a new password` (followed by the `New password:` prompt)	This message occurs the first time a user logs in and chooses an initial password to access the account.
`Couldn't fork a process!`	This message occurs when the server could not fork a child process during login. The most common cause of this message is that the system has reached its maximum number of processes. You can either kill some unneeded processes (if you are already logged into that system as `root`) or increase the number of processes your system can handle.

Login Problems in the CDE

Problems associated with logging into the CDE range from a user being unable to login (and returning to the CDE login screen), to the custom environment not loading properly. In general, the system does not return error messages to the user from the CDE. The following is a list of files and directories that provide troubleshooting information about the CDE:

- `/usr/dt/bin/Xsession`

  This file is the configuration script for the login manager. This file should not be edited. The first user-specific file that the `Xsession` script calls is the `$HOME/.dtprofile` file.

- `$HOME/.dtprofile`

  By default, the file does not contain much content, except for examples. It contains a few `echo` statements for session logging purposes, and the `DTSOURCEPROFILE` variable is set. But it also contains information about how it might be edited. The user can edit this file to add user-specific environment variables.

- `DTSOURCEPROFILE=true`

  This line allows the user's `$HOME/.login` file (for `csh` users) or the `$HOME/.profile` (for other shell users) to be sourced as part of the startup process.

  Sometimes a `.login` or `.profile` file contains problem commands that cause the shell to crash. If the `.dtprofile` file is set to source a `.login` or `.profile` file that has problem commands, desktop startup might fail.

  Consequently, no desktop appears. Instead, the system redisplays the Solaris OE CDE login screen. Startup errors from the `.login` or `.profile` file are usually noted in the `$HOME/.dt/startlog` file. Use a Failsafe login Session or a command-line login to debug problem commands in the `.login` or `.profile` files.

- `$HOME/.dt/sessions`

  This directory structure contains files and directories that configure the display of the user's custom desktop and determine the applications that start when the user logs in. Look for recent changes to files and for changes to the directory structure. For example, examine the `home` directory and the `home.old` directory or a `current` directory and the `current.old` directory. Compare the changes. The changes could provide information on a new application or on changes in the saved desktop that cause the user's login to fail.

- `$HOME/.dt`

  Upon removing the entire `.dt` directory structure, log out, and log back in again for the system to rebuild a default `.dt` file structure. This action allows the user to get back into the system if the problem with the CDE files cannot be resolved.

Table 10-14 shows the locations of and information found in error logs for the CDE.

Table 10-14 DE Error Log Locations

Location	Error Log
/var/dt/Xerrors	The Solaris OE CDE login window system errors that occur prior to user login
$HOME/.dt/startlog	The Solaris OE CDE errors that occur during the startup of the Xsession script, while processing the .dtprofile, .login, or .profile file
$HOME/.dt/errorlog.old $HOME/.dt/errorlog.older	The Solaris OE CDE errors that occur after the Xsession script start up.
$HOME/.dt/sessionlogs	Directory of session logs for Session Manager and Window Manager errors

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Adding User Accounts and Group Entries (Level 1)

In this exercise, you use the Solaris Management Console, as well as the smuser, smgroup, usermod, userdel, groupadd, and groupdel commands, to create, modify, and delete multiple user accounts and group entries.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed. Refer to Table 10-15 and Table 10-16 as needed.

Table 10-15 Group Specifications

Group Name	GID Number
class1	101
class2	102

Table 10-16 User Specifications

User Name	Password	Shell	UID	Primary Group	Secondary Group
user3	123pass	Korn	1003	10	class1
user4	123pass	C	1004	10	class1
user5	123pass	Bourne	1005	10	
locked1	Select Account is Locked	Bourne	2001	10	
cleared1	Select User must set password at next login	Bourne	2002	10	

Tasks

Complete the following tasks:

- Disable the Solaris OE registration window.

  (Steps 1–5 of Task 1 in the Level 2 lab)

- Working from Table 10-15 and Table 10-16 on page 10-37, create two new groups and two new users by using the groupadd, smgroup, useradd, and smuser commands.

  (Steps 1–2 of Tasks 2 and 3 in the Level 2 lab)

- Launch the Solaris Management Console, and create a user template to add users that do not use automounted home directories.

  (Step 3 of Task 3 in the Level 2 lab)

- Using the Solaris Management Console, add the new users user5, locked1, and cleared1 with characteristics from Table 10-16 on page 10-37.

  (Steps 4–5 of Task 3 in the Level 2 lab)

- Verify that the shells you specify are set in the /etc/passwd file. Determine if the password strings for users with the same password are also the same in the /etc/shadow file. Check the password strings for the users locked1 and cleared1. Verify that the users user3 and user4 are secondary members of the class1 group.

  (Steps 1–4 of Task 4 in the Level 2 lab)

- Determine what happens when you try to log in as the user locked1. Verify that you can log in as the user cleared1. Record the password requirements indicated.

  (Steps 5–6 of Task 4 in the Level 2 lab)

- Establish password aging for the user user5. Determine what happens when you attempt to log in as that user. Log in as user5 and attempt to change the password from the command line. Log in as the root user when you are finished.

  (Steps 1–4 of Task 5 in the Level 2 lab)

- Use the groupadd command to add a group called class3. Use the usermod command to change the UID number, home directory, and user name for the user locked1. Verify that the changes exist in the /etc/passwd file.

  (Steps 1–2 of Task 5 in the Level 2 lab)

- Use the usermod command to change the login shell of user5 to ksh. Use the userdel command to delete the user user3. Verify that the home directory has been deleted. Use the groupmod command to rename the group class1 to group1. Use the groupdel command to remove the group class2. Verify the changes to the /etc/group file.

  (Steps 3–7 of Task 5 in the Level 2 lab)

Exercise: Adding User Accounts and Group Entries (Level 2)

In this exercise, you use the Solaris Management Console, as well as the `smuser`, `smgroup`, `usermod`, `userdel`, `groupadd`, and `groupdel` commands, to create, modify, and delete multiple user accounts and group entries.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed. Refer to Table 10-15 and Table 10-16 on page 10-37 as needed.

Task Summary

In this exercise, you accomplish the following:

- Disable the Solaris OE registration window.
- Working from Table 10-15 and Table 10-16 on page 10-37, create two new groups and two new users by using the commands `groupadd`, `smgroup`, `useradd`, and `smuser`.
- Launch the Solaris Management Console, and create a user template to add users that do not use automounted home directories.
- Using the Solaris Management Console, add the new users `user5`, `locked1` and `cleared1` with characteristics from Table 10-16 on page 10-37.
- Verify that the shells you specify are set in the `/etc/passwd` file. Determine if the password strings for users with the same password are also the same in the `/etc/shadow` file. Check the password strings for the users `locked1` and `cleared1`. Verify that the users `user3` and `user4` are secondary members of the `class1` group.
- Determine what happens when you try to log in as the user `locked1`. Verify that you can log in as the user `cleared1`. Record the password requirements indicated.
- Establish password aging for `user5`. Determine what happens when you attempt to log in as that user. Log in as `user5` and attempt to change the password from the command line. Log in as the `root` user when you are finished.

- Use the `groupadd` command to add a group called `class3`. Use the `usermod` command to change the UID number, home directory, and user name for the user `locked1`. Verify that the changes exist in the `/etc/passwd` file.
- Use the `smuser` command to change the login shell of `user5` to Ksh. Use the `userdel` command to delete the user `user3`. Verify that the user's home directory has been deleted. Use the `smgroup` command to rename the group `class1` to `group1`. Use the `groupdel` command to remove the group `class2`. Verify the changes to the `/etc/group` file.

Tasks

Complete the following tasks.

Task 1 – Disabling the Solaris OE Registration Window

Complete the following steps:

1. Disable the Solaris OE Registration window so that it does not appear whenever a new user logs in from the CDE.
2. Log in as the `root` user (or use the `su` command to change to the `root` user).
3. Change to the `/etc/default` directory.
4. In the `default` directory, create the `solregis` file.

#vi solregis

5. In the `solregis` file, type the keyword `DISABLE=1` (note that the character "`1`" is the number one).
6. Save this file, and exit the editor.

Task 2 – Adding Group Entries

Complete the following steps:

Note – Refer to Table 10-15 on page 10-37 for details while adding groups.

1. As the `root` user, open a terminal window.
2. Add the two groups `class1` and `class2` with the `groupadd` and `smgroup` commands, respectively.

Task 3 – Adding User Accounts

Complete the following steps:

Note – *Refer to Table 10-16 on page 10-37 for details while adding users with the various tools.*

---

1. Add a user named user3 by using the useradd command.
2. Add a user named user4 by using the smuser command.
3. Launch the Solaris Management Console by typing smc on the command line. After the Solaris Management Console appears, create a user template to add user accounts that do not use automounted home directories by performing the following:
   a. Select This Computer, and then select System Configuration. Then select Users, and then select User Templates to open the User Templates tool.
   b. From the Menu Bar, select Action. Then select Add User Template.
   c. The Add User Template window appears, containing blank fields for a template name and description. Enter the name 239user in the User Template Name field, and SA239 for the Description field.
   d. Click the Home Directory Tab and uncheck the Automatically Mount Home Directory check box. Enter the name of your system in the Home Directory Server field.
   e. Click OK to create your template.
4. Click User Accounts, and add the user5 account by selecting Action, then selecting Add User, and then selecting From Template on the menu bar.

   The Add User From Template window appears. Enter user5 in the User Name field, and select 1005 as the User ID Number. For the password, click User Must Use, and enter 123pass in both password fields. Click OK.
5. From the Solaris Management Console, add additional users locked1 and cleared1 by using the 239user template. While adding the cleared1 user, select the password option User Must Set Password At Next Login. After adding both users, double-click the locked1 user and select the tab General. Under the Account Availability section, select the button Account is Locked. Also select the shell as listed in Table 10-16 on page 10-37.

Task 4 – Examining Configuration Files

Complete the following steps:

1. Examine the contents of the /etc/passwd file. What are the full path names of the shells used by user3, user4, and user5?
2. Examine the contents of the /etc/shadow file. What text is found in the password field for the users locked1 and cleared1?
3. You used the same password for user3 through user5. Are the password strings the same in the /etc/shadow file?
4. Examine the contents of the /etc/group file. Verify that user3 and user4 are both listed as secondary members of the class1 group. Are they?
5. Log out of the CDE, and attempt to log in as locked1. Are you able to log in?
6. Attempt to log in as cleared1. What happens? Attempt to use the password abcdefg. What are the system requirements for the password?

   Use the password abc123. Log in as cleared1 after you establish a password to verify that the login works. Log out, and log in as the root user.

Task 5 – Establishing Password Aging

Complete the following steps:

1. Start the Solaris Management Console, and go back into the User Accounts Tool. Select user5 from the list of users. Change the password options information for user5 so that it matches the following information. Click OK when you are finished, and exit the Solaris Management Console.

User Must Keep For:	1 (one day)
Before Change Alert User:	1 (one day)
User Must Change Within:	2 (two days)
Expires If Not Used For:	1 (one day)

2. Log out of your root login session. Attempt to log in as user5. What happens? Supply a new password if necessary.

3. Complete the login as user5. Open a terminal window, and attempt to change the password you just set. What happens?
4. Log out, and log in again as the root user.

Task 6 – Modifying User Accounts and Group Entries

Complete the following steps:

1. Use the groupadd command to create a new group entry called class3 that uses GID number 103.
2. Use the usermod command to change the login name of locked1 to user6, the UID to 3001, and the home directory of locked1 to user6. Verify that the changes you request are recorded in the /etc/passwd file and the directory that was moved.
3. Use the smuser modify command to change the login shell of user5 to /bin/ksh. Verify that the changes you request are recorded in the /etc/passwd file.
4. Use the userdel command to delete the user account cleared1 and the related home directory. Verify that the /export/home/cleared1 directory no longer exists.
5. Use the smgroup command to change the group name of class1 to group1.
6. Use the groupdel command to remove the group entry class2.
7. Verify that the commands used to modify group entries have correctly modified the /etc/group file.

Exercise: Adding User Accounts and Group Entries (Level 3)

In this exercise, you use the Solaris Management Console, as well as the `smuser`, `smgroup`, `usermod`, `userdel`, `groupadd`, `groupmod`, and `groupdel` commands, to create, modify, and delete multiple user accounts and group entries.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed. Refer to Table 10-15 and Table 10-16 on page 10-37 as needed.

Note – Some of the commands displayed in this section are quite long and will wrap to the next line. You should consider all of the bold typeface commands that follow a command line prompt to be all one line.

Task Summary

In this exercise, you accomplish the following:

- Disable the Solaris OE registration window.
- Working from Table 10-15 and Table 10-16 on page 10-37, create two new groups and two new user accounts using the commands `groupadd`, `smgroup`, `useradd`, and `smuser`.
- Launch the Solaris Management Console and create a user template to add users that do not use automounted home directories.
- Using the Solaris Management Console, add the new user accounts `user5`, `locked1`, and `cleared1` with characteristics from Table 10-16 on page 10-37.
- Verify that the shells you specify are set in the `/etc/passwd` file. Determine if the password strings for users with the same password are also the same in the `/etc/shadow` file. Check the password strings for the users `locked1` and `cleared1`. Verify that the users `user3` and `user4` are secondary members of the `class1` group.
- Determine what happens when you try to log in as the user `locked1`. Verify that you can log in as the user `cleared1`. Record the password requirements indicated.

- Establish password aging for the user user5. Determine what happens when you attempt to log in as that user. Log in as user5 and attempt to change the password from the command line. Log in as root when you are finished.
- Use the groupadd command and to add a group called class3. Use the usermod command to change the UID number, home directory, and user name for the user locked1. Verify that the changes exist in the /etc/passwd file.
- Use the usermod command to change the login shell of user5 to ksh. Use the userdel command to delete the user3 account. Verify that the user's home directory has been deleted. Use the groupmod command to rename the group class1 to group1. Use the groupdel command to remove the group class2. Verify the changes to the /etc/group file.

Tasks and Solutions

Complete the following tasks.

Task 1 – Disabling the Solaris OE Registration Window

Complete the following steps:

1. Disable the Solaris OE Registration window so that it does not appear whenever a new user logs in from the CDE.
2. Log in as the root user (or use the su command to change the root user).
3. Change to the /etc/default directory.
4. In the default directory, create the file solregis.

#vi solregis

5. In the solregis file, type the keyword DISABLE=1 (note that the character "1" is the number one).
6. Save this file, and exit the editor.

Task 2 – Adding Group Entries

Complete the following steps:

Note – Refer to Table 10-15 on page 10-37 for details while adding groups.

---

1. As the root user, open a terminal window.
2. Add the two groups class1 and class2, with groupadd and smgroup commands, respectively.

```
# groupadd -g 101 class1
# /usr/sadm/bin/smgroup add -- -n class2 -g 102
```

Task 3 – Adding User Accounts

Complete the following steps:

Note – Refer to Table 10-16 on page 10-37 for details while adding users with the various tools.

---

1. Add a user named user3 by using the useradd command.

```
# useradd -u 1003 -g 10 -G class1 -d /export/home/user3 -m -s /bin/ksh
user3
# passwd user3
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for user3
```

2. Add a user named user4 by using the smuser command.

```
# /usr/sadm/bin/smuser add -- -n user4 -u 1004 -g 10 -G class1 -d
/export/home/user4 -s /bin/csh -x autohome=N
# passwd user4
New Password: 123pass
Re-enter new Password: 123pass
passwd: password successfully changed for user4
```

3. Launch the Solaris Management Console by typing smc& on the command line. After the Solaris Management Console appears, create a user template to add user accounts that do not use automounted home directories by performing the following:
   a. Select This Computer, and then select System Configuration. Then select Users, and then select User Templates to open the User Templates tool.

b. From the Menu Bar, select Action, and then select Add User Template.

c. The Add User Template window appears, containing blank fields for a template name and description. Enter the name 239user in the User Template Name field, and SA239 for the Description field.

d. Click the Home Directory Tab and uncheck the Automatically Mount Home Directory check box. Enter the name of your system in the Home Directory Server field.

e. Click OK to create your template.

4. Click User Accounts, and add the user5 account by selecting Action, then selecting Add User, and then selecting From Template on the menu bar.

   The Add User From Template window appears. Enter user5 in the User Name field and select 1005 as the UID Number. For password, click the button called User Must Use, and enter 123pass in both password fields. Click OK.

5. From the Solaris Management Console, add the users locked1 and cleared1 by using the 239user template. While adding the cleared1 user, select the password option User Must Set Password At Next Login. After adding both users, double-click the locked1 user and select the tab General. Under the Account Availability section, select Account is Locked. Also select the shell as listed in Table 10-16 on page 10-37.

Task 4 – Examining Configuration Files

Complete the following steps:

1. Examine the contents of the /etc/passwd file. What are the full path names of the shells used by user3, user4, and user5?

user3	/bin/ksh
user4	/bin/csh
user5	/bin/sh

2. Examine the contents of the /etc/shadow file. What text is found in the password field for the users locked1 and cleared1?

locked1	*LK*
cleared1	none

3. You used the same password for user3 through user5. Are the password strings the same in the /etc/shadow file?

   No.

4. Examine the contents of the /etc/group file. Verify that user3 and user4 are both listed as secondary members of the class1 group. Are they?

   The names user3 *and* user4 *should be listed in the last field for the* class1 *group.*

5. Log out of the CDE, and attempt to log in as locked1. Are you able to log in?

   No, you get a message that says login incorrect, *no matter what you use as a password.*

6. Attempt to log in as cleared1. What happens? Attempt to use the password abcdefg. What are the system requirements for the password? You must not press Return when you are asked for an initial password.

   You must choose an initial password for this user and then log in again. The first six characters must contain at least two alphabetic characters and at least one numeric or special character.

   Use the password abc123. Log in as cleared1 after you establish a password to verify that the login works. Log out, and log in as the root user.

Task 5 – Establishing Password Aging

Complete the following steps:

1. Start the Solaris Management Console, and go back into the User Accounts tool. Select user5 from the list of users. Change the password options information for user5 so that it matches the following information. Click OK when you are finished, and exit the Solaris Management Console.

User Must Keep For:	1 (one day)
Before Change Alert User:	1 (one day)
User Must Change Within:	2 (two days)
Expires If Not Used For:	1 (one day)

2. Log out of your root login session. Attempt to log in as user5. What happens? Supply a new password if necessary.

   You must supply a new password before you can log in.

3. Complete the login as user5. Open a terminal window, and attempt to change the password you just set. What happens?

   When you log in, a warning indicates that your password expires in two days.

   When you try to change your password, the following error message appears:

   ```
   passwd: Sorry: less than 1 days since the last change.
   Permission denied
   ```

4. Log out, and log in again as the root user.

Task 6 – Modifying User Accounts and Group Entries

Complete the following steps:

1. Use the `groupadd` command to create a new group entry called `class3` that uses GID number 103.

```
# groupadd -g 103 class3
```

2. Use the `usermod` command to change the login name of `locked1` to `user6`, the UID to `3001`, and the home directory of `locked1` to `user6`. Verify that the changes you request are recorded in the `/etc/passwd` file and that the directory was moved.

```
# usermod -u 3001 -d /export/home/user6 -m -l user6 locked1
```

The `/etc/passwd` file should reflect the new UID number and user name. The directory under `/export/home` should be renamed.

3. Use the `smuser modify` command to change the login shell of `user5` to `/bin/ksh`. Verify that the changes you request are recorded in the `/etc/passwd` file.

```
# /usr/sadm/bin/smuser modify -- -n user5 -s /bin/ksh
```

The `/etc/passwd` file should show that the shell is `/bin/ksh`.

4. Use the `userdel` command to delete the user account `cleared1` and the related home directory. Verify that the `/export/home/cleared1` directory no longer exists.

```
# userdel -r cleared1
```

The `/export/home/cleared1` directory should no longer exist.

5. Use the `smgroup` command to change the group name of `class1` to `group1`.

```
# /usr/sadm/bin/smgroup modify -- -n class1 -N group1
```

6. Use the `groupdel` command to remove the group entry `class2`.

```
# groupdel class2
```

7. Verify that the commands used to modify group entries have correctly modified the `/etc/group` file.

The group `group1` should exist. The groups `class1` and `class2` should not exist.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Managing Initialization Files

The environment maintained by the shell includes variables that are defined by the `login` program, the system initialization files, and the user initialization files.

When users log in to the system, their login shells look for and execute two different types of initialization files. The first type controls the system-wide environment. The second type controls the user's environment. The six shells available in the Solaris 9 OE provide basic features and a set of variables which the `root` user or a regular user can set in the initialization files to customize the shell environment.

The shells support two types of variables:

- Environment variables – Variables that provide information about the user's environment to every shell program that is started.
- Local variables – Variables that affect only the current shell. Any subshell started would not have knowledge of these variables.

Introducing System-Wide Initialization Files

As the system administrator, you maintain the system-wide initialization files. These files provide an environment for the entire community of users who log in to the system. The Solaris OE provides the system initialization files. They reside in the `/etc` directory.

The `/etc/profile` file and the `/etc/.login` file are the two main system initialization files.

The Bourne, Korn, and BASH login shells look for and execute the system initialization file `/etc/profile` during login.

The C login shell looks for and executes the system initialization file `/etc/.login` during the login process.

There are no default global initialization files for the Z or TC shells.

Note – The default files /etc/profile and /etc/.login check disk usage quotas, print the message of the day from the /etc/motd file, and check for mail. None of the messages are printed to the screen if the .hushlogin file exists in the user's home directory.

Introducing User Initialization Files

As the system administrator, you set up the user initialization files that are placed in each user account's home directory when the user is created.

The primary purpose of the user initialization files is to define the characteristics of a user's work environment, such as the command-line prompt, the environment variables, and the windowing environment.

Only the owners of the files or the root user can change or customize the content of these files.

Table 10-17 shows the initialization files necessary for each primary shell available in the Solaris 9 OE.

Table 10-17 Initialization Files for the Primary Shells

Shells	System-Wide Initialization Files	Primary User Initialization Files Read at Login	User Initialization Files Read When a New Shell Is Start	Shell Path Name
Bourne	/etc/profile	$HOME/.profile		/bin/sh
Korn	/etc/profile	$HOME/.profile $HOME/.kshrc	$HOME/.kshrc	/bin/ksh
C	/etc/.login	$HOME/.cshrc $HOME/.login	$HOME/.cshrc	/bin/csh

For additional information about the Z, BASH, and TC shells available in the Solaris 9 OE, refer to the online manual pages.

Note – By default, the root user's login shell is the Bourne shell, and the shell entry in the /etc/passwd file appears as /sbin/sh.

When a user logs in to the system, the system invokes the user's login shell program. The shell program looks for its initialization files in a specific order, executes the commands contained in each file, and displays the shell prompt on the user's screen.

Customizing the User's Work Environment

The Solaris OE provides a set of initialization file templates. The /etc/skel directory contains the initialization file templates. Table 10-18 shows the default initialization file templates and the user initialization files for the Bourne, Korn, and C shells.

Table 10-18 Default User Initialization Files

Shell	Initialization File Templates	User Initialization Files
Bourne	/etc/skel/local.profile	$HOME/.profile
Korn	/etc/skel/local.profile	$HOME/.profile
C	/etc/skel/local.cshrc /etc/skel/local.login	$HOME/.cshrc $HOME/.login

Note – The useradd command copies files from the /etc/skel directory to the $HOME directory. The smuser command copies files from the /etc/skel directory to the $HOME directory and renames them to the appropriate file names.

The root user can customize these templates to create a standard set of user initialization files. A standard set of user initialization files provides a common work environment for each user. When the root user creates new user accounts, some or all of these initialization files are automatically copied to each new user's home directory.

Users can then edit their initialization files to further customize their environments for each shell.

Table 10-19 shows some of the variables available for customizing a user's shell environment.

Table 10-19 Login Variables

Variable Name	Set By	Description
LOGNAME	Login	Defines the user's login name.
HOME	Login	Sets the path to the user's home directory. It is the default argument for the cd command.
SHELL	Login	Sets the path to the default shell.
PATH	Login	Sets the default path that the shell searches to find commands.
MAIL	Login	Sets the path to the user's mailbox.
TERM	Login	Defines the terminal.
LPDEST	Not set by default	Sets the user's default printer.
PWD	Shell	Defines the current working directory.
PS1	Shell	Defines the shell prompt for the Bourne or Korn shell.
prompt	Shell	Defines the shell prompt for the C shell.

Note – For complete information on all variables used by the default shells, see the following man pages: sh(1), ksh(1), csh(1), zsh(1), bash(1), and tcsh(1).

A user can change the values of the predefined variables and specify additional variables.

Table 10-20 shows how to set environment variables in the user initialization files of the Bourne, Korn, and C shells.

Table 10-20 Setting Environment Variables

Shell	User's Initialization File

Bourne or Korn	`VARIABLE=value ; export VARIABLE`
	For example:
	`PS1="$HOSTNAME "; export PS1`

C	`setenv variable value`
	For example:
	`setenv LPDEST laserprinter`

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Modifying Initialization Files (Level 1)

In this exercise, complete the following tasks:

- Modify initialization file templates in the /etc/skel directory
- Create user accounts that use the initialization files

Preparation

This exercise requires the skills practiced in the previous exercise. The user accounts that you create in this exercise are required in later sections of the course. Refer to the lecture notes as necessary to perform the tasks listed.

Tasks

Complete the following tasks:

- Modify the template for Bourne shell users. Set the EDITOR to vi, LPDEST to printer1, EXINIT to set showmode autoindent and number, and ENV to source the .kshrc file.

  (Steps 1–3 in the Level 2 lab)
- Use the Solaris Management Console to create a new user account called user9 that uses the Korn shell. Log in as the new user, and verify that all the variables you set in local.profile are set correctly in the user's environment.

  (Steps 4–6 in the Level 2 lab)
- Create a .kshrc file for the new user account that includes two aliases and sets the primary prompt to echo the current working directory. Log out, and log in again as the same user to verify that the .kshrc file works. Log out, and log in again as the root user.

  (Steps 7–9 in the Level 2 lab)
- Use the useradd command to create a new user account called user10 that uses the Korn shell. Log in as this user, and record the list of initialization files in the home directory. Copy the appropriate file to the .profile file. Test the login to verify that the list of variables is set the same as those of the first user you created. Log out, and log in as the root user when you are finished.

  (Steps 9–13 in the Level 2 lab)

Exercise: Modifying Initialization Files (Level 2)

In this exercise, complete the following tasks:

- Modify initialization file templates in the /etc/skel directory
- Create user accounts that use the initialization files

Preparation

This exercise requires the skills practiced in the previous exercise. The user accounts that you create in this exercise are required in later sections of the course. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Modify the template for Bourne shell users. Set the EDITOR to vi, LPDEST to printer1, EXINIT to set showmode autoindent and number, and ENV to source the .kshrc file.
- Use the Solaris Management Console to create a new user account called user9 that uses the Korn shell. Log in as the new user, and verify that all the variables you set in local.profile are set correctly in the user's environment.
- Create a .kshrc file for the new user account that includes two aliases and sets the primary prompt to echo the current working directory. Log out, and log in again as the same user to verify that the .kshrc file works. Log out, and log in again as the root user.
- Use the useradd command to create a new user account called user10 that uses the Korn shell. Log in as this user, and record the list of initialization files in the home directory. Copy the appropriate file to the .profile file. Test the login to verify that the list of variables is set the same as those of the first user you created. Log out, and log in as the root user when you are finished.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window.
2. Change to the /etc/skel directory.
3. Use the vi editor to edit the local.profile file, and make the following changes:
   a. Edit the line that declares the PATH variable so that it reads as follows. Enter this text as one line (no spaces).

```
PATH=/usr/sbin:/sbin:/usr/sadm/bin:/usr/dt/bin:/usr/openwin/bin:/usr/bin:
/usr/ccs:.
```

   b. Add the following lines below the PATH variable you just edited:

```
EDITOR=vi
LPDEST=printer1
EXINIT='set showmode autoindent number'
ENV=$HOME/.kshrc
```

   c. Change the line that reads:

```
export PATH
```

   so that it reads:

```
export PATH EDITOR LPDEST EXINIT ENV
```

4. Use the Solaris Management Console to create a new user account with the following characteristics. Exit the Solaris Management Console when you are finished.

User Name:	user9
User ID:	1009
Primary Group:	staff
Login Shell:	Korn
Password:	123pass

5. Log out, and log in again as user9. Open a terminal window.
6. Verify that the PATH, LPDEST, EDITOR, EXINIT, and ENV variables are set according to the changes you made in the /etc/skel/local.profile file.

   Do they match?

7. Create a file called .kshrc in user9's home directory.

   Insert the following lines. A space follows the PWD in the last line.

```
set -o noclobber
set -o ignoreeof
alias h=history
alias c=clear
PS1='$PWD$ '
```

8. Log out, and then log in again as user9. Open a terminal window, and verify that your new variables work.

   Do they work?

9. Log out, and log in again as the root user. Use the useradd command to create a new user account called user10 with the following characteristics:

User Name:	user10
User ID:	1010
Primary Group:	10
Login Shell:	Korn
Home Directory:	/export/home/user10
Comment:	SA-239 Student
Password:	cangetin

10. Log out, and log in again as user10. Open a terminal window. What shell initialization files exist in your home directory?

    Which of these are the same as /etc/skel/local.profile?

11. Copy the local.profile file to the .profile file.

12. Log out, and log in again as user10. Verify that the variables set for the user9 login are also set for this login.

    Do they match?

13. Log out, and log in again as the root user.

Exercise: Modifying Initialization Files (Level 3)

In this exercise, complete the following tasks:

- Modify initialization file templates in the /etc/skel directory
- Create user accounts that use the initialization files

Preparation

This exercise requires the skills practiced in the previous exercise. The user accounts that you create in this exercise are required in later sections of the course. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Edit the /etc/skel/local.profile file so that it sets the PATH variable to a specific list of directories. Set the EDITOR, LPDEST, EXINIT, and ENV variables to appropriate values.
- Use the Solaris Management Console to create a new user account called user9 that uses the Korn shell. Log in as the new user, and verify that all the variables you set in local.profile are set correctly in the user's environment.
- Create a .kshrc file for the new user account that includes two aliases and sets the primary prompt to echo the current working directory. Log out, and log in again as the same user to verify that the .kshrc file works. Log out, and log in again as the root user.
- Use the useradd command to create a new user account called user10 that uses the Korn shell. Log in as this user, and record the list of initialization files in the home directory. Copy the appropriate file to the .profile file. Test the login to verify that the list of variables is set the same as those of the first user you created. Log out, and log in as the root user when you are finished.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window.
2. Change to the /etc/skel directory.

```
# cd /etc/skel
```

3. Use the vi editor to edit the local.profile file, and make the following changes:

```
# vi local.profile
```

   a. Edit the line that declares the PATH variable so that it reads as follows. Enter this text as one line (no spaces).

```
PATH=/usr/sbin:/sbin:/usr/sadm/bin:/usr/dt/bin:/usr/openwin/bin:/usr/bin:
/usr/ucb:.
```

   b. Add the following lines below the PATH variable you just edited:

```
EDITOR=vi
LPDEST=printer1
EXINIT='set showmode autoindent number'
ENV=$HOME/.kshrc
```

   c. Change the line that reads:

```
export PATH
```

   so that it reads:

```
export PATH EDITOR LPDEST EXINIT ENV
```

4. Use the Solaris Management Console to create a new user with the following characteristics. Exit the Solaris Management Console when you are finished.

User Name:	user9
User ID:	1009
Primary Group:	staff
Login Shell:	Korn
Password:	123pass

5. Log out, and log in again as user9. Open a terminal window.

6. Verify that the PATH, LPDEST, EDITOR, EXINIT, and ENV variables are set according to the changes you made in the `/etc/skel/local.profile` file.

```
$ echo $PATH
$ echo $LPDEST
$ echo $EDITOR
$ echo $EXINIT
$ echo $ENV
```

Do they match?

These variables should match the settings made in the `local.profile` *file.*

7. Create a file called `.kshrc` in `user9`'s home directory.

```
$ cd
$ vi .kshrc
```

Insert the following lines. A space follows the `$PWD$` in the last line.

```
set -o noclobber
set -o ignoreeof
alias h=history
alias c=clear
PS1='$PWD$ '
```

8. Log out, and then log in again as `user9`. Open a terminal window, and verify that your new variables work.

```
$ cd /tmp
$ cd
$ c
$ h
```

Do they work?

These variables should function according to the values set in `.kshrc`. *The prompt should reflect your current directory, and the aliases should clear the screen and present a history list.*

9. Log out, and log in again as the root user. Use the useradd command to create a new user account called user10 with the following characteristics:

User Name:	user10
User ID:	1010
Primary Group:	10
Login Shell:	Korn
Home Directory:	/export/home/user10
Comment:	SA-239 Student
Password:	cangetin

```
# useradd -u 1010 -g 10 -d /export/home/user10 -m -s /bin/ksh -c "SA-239
Student" user10
64 blocks
# passwd user10
New password: cangetin
Re-enter new password: cangetin
```

10. Log out, and log in again as user10. Open a terminal window. What shell initialization files exist in your home directory?

```
$ ls -la
.profile, local.profile, local.login, local.cshrc
```

Which of these are the same as the /etc/skel/local.profile file?

The local.profile *file.*

11. Copy the local.profile file to the .profile file.

```
$ cp local.profile .profile
```

12. Log out, and log in again as user10. Verify that the variables set for the user3 login are also set for this login.

```
$ echo $PATH
$ echo $LPDEST
$ echo $EDITOR
$ echo $EXINIT
$ echo $ENV
```

Do they match?

These variables should match the settings made in the local.profile *file.*

13. Log out, and log in again as the root user.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 11

Performing System Security

Objectives

Upon completion of this module, you should be able to:

- Monitor system access
- Switch users on a system
- Control system access
- Restrict access to data in files

The following course map shows how this module fits into the current instructional goal.

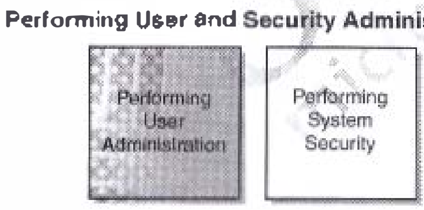

Figure 11-1 Course Map

Monitoring System Access

All systems should be monitored routinely for unauthorized user access. You can determine who is or who has been logged into the system by executing commands and examining log files.

Displaying Users on the Local System

The who command displays a list of users currently logged in to the local system. It displays each user's login name, the login device (TTY port), the login date and time. The command reads the binary file /var/adm/utmpx to obtain this information and information about where the users logged in from.

If a user is logged in remotely, the who command displays the remote host name, or Internet Protocol (IP) address in the last column of the output.

```
# who
user2      pts/2       Feb  7 13:53    (sys43)
root       pts/5       Feb  6 09:22    (129.147.4.13)
root       pts/3       Feb  7 14:27    (sys41)
root       console     Feb  5 11:05    (:0)
#
```

The second field displayed by the who command defines the user's login device, which is one of the following:

- console – The device used to display system boot and error messages
- pts – The pseudo device that represents a login or window session without a physical device
- term – The device physically connected to a serial port, such as a terminal or a modem

Note – The who command has many options, one of which is the -m option. The who -m command outputs information about only the current terminal window.

Displaying Users on Remote Systems

The rusers command produces output similar to that of the who command, but it displays a list of the users logged in on local and remote hosts. The list displays the user's name and the host's name in the order in which the responses are received from the hosts.

A remote host responds only to the rusers command if its rpc.rusersd daemon is enabled. The rpc.rusersd daemon is the network server daemon that returns the list of users on the remote hosts.

Note – The full path to this network server daemon is /usr/lib/netsvc/rusers/rpc.rusersd.

The following is the command format for the rusers command:

```
rusers -options hostname
```

The rusers -l command displays a long list of the login names of users who are logged in on local and remote systems. The output displays the name of the system into which a user is logged, the login device (TTY port), the login date and time, the idle time, and the login host name. If the user is not idle, no time is displayed in the idle time field. The term idle means that the user is not actively doing anything at the time on the terminal, which would denote the user is probably at screen lock or away from the terminal.

The following is an example of the rusers command:

```
# rusers -l
Sending broadcast for rusersd protocol version 3...
root        instructor:console    Feb  5 11:05   50:15 (:0)
root        instructor:pts/5      Feb  6 09:22   27:38 (129.147.4.13)
root        instructor:pts/6      Feb  4 13:36    5:08 (129.147.48.219)
root        instructor:pts/7      Feb  4 13:36   25:50 (129.147.48.219)
root        instructor:pts/2      Feb  6 09:23   27:10 (129.147.4.13)
root        instructor:pts/10     Feb  7 07:35      45 (lightbandit)
root        instructor:pts/12     Feb  7 09:38      44 (lightbandit)
root        instructor:pts/11     Feb  7 14:16         (129.147.4.20)
user2       sys44:pts/2           Feb  7 13:53      45 (instructor)
root        sys11ext:console      Feb  6 13:17   23:52
user2       sys41:pts/1           Feb  7 13:45      44 (instructor)
root        sys41:pts/3           Feb  7 14:32         (instructor)
Sending broadcast for rusersd protocol version 2...
```

Displaying User Information

To display detailed information about user activity that is either local or remote, use the finger command.

The finger command displays:

- The user's login name
- The home directory path
- The login time
- The login device name
- The data contained in the comment field of the /etc/passwd file (usually the user's full name)
- The login shell
- The name of the host, if the user is logged in remotely, and any idle time

The following is the command format for the finger command:

```
finger [-bfhilmpqsw] [username...]
finger [-l] [ username@hostname1 [ @hostname ]]
```

The -m option matches arguments only on username (not the first or last name that might appear in the comment field of /etc/passwd).

To display information for usera, perform the command:

```
# finger -m usera
Login name: usera                In real life: Alpha User located in Office #4
Directory: /home/usera           Shell: /bin/sh
On since Dec 17 10:32:53 on console from :0
1 minute 47 seconds Idle Time
No unread mail
No Plan.
```

If users create the standard ASCII files .plan or .project in their home directories, the content of those files is shown as part of the output of the finger command.

These files are traditionally used to outline a user's current plans or projects and must be created with file access permissions set to 644 (rw-r--r--).

Note – You get a response from the finger command only if the in.fingerd daemon is enabled.

Displaying a Record of Login Activity

Use the last command to display a record of all logins and logouts with the most recent activity at the top of the output. The last command reads the binary file /var/adm/wtmpx, which records all logins, logouts, and reboots.

Each entry includes the user name, the login device, the host that the user is logged in from, the date and time that the user logged in, the time of logout, and the total login time in hours and minutes, including entries for system reboot times.

The output of the last command can be extremely long. Therefore, you might want to use it with the -n *number* option to specify the number of lines to display.

The following is an example of the last command:

```
# last
user9    console      :0               Mon Dec 17 10:38   still logged in
root     pts/4        129.147.4.12     Mon Dec 17 10:33   still logged in
usera    console      :0               Mon Dec 17 10:32 - 10:38  (00:05)
reboot   system boot                   Fri Dec 14 09:58
(output truncated)
```

You can use the last command also to display information about an individual user if you supply the user's login name as an argument.

```
# last user9
user9     console     :0               Mon Dec 17 10:38   still logged in
user9     console     :0               Fri Dec 14 10:18 - 10:25   (00:07)
(output truncated)
```

To view the last five system reboot times only, perform the command:

```
# last -n 5 reboot
reboot     system boot                 Wed Feb 20 13:20
reboot     system boot                 Wed Feb 20 13:18
reboot     system boot                 Fri Feb  1 12:46
reboot     system boot                 Thu Jan 17 09:02
reboot     system boot                 Thu Jan 17 08:55
```

Recording Failed Login Attempts

When a user logs in to a system either locally or remotely, the login program consults the /etc/passwd and the /etc/shadow files to authenticate the user. It verifies the user name and password entered.

If the user provides a login name that is in the /etc/passwd file and the correct password for that login name, the login program grants access to the system.

If the login name is not in the /etc/passwd file or the password is not correct for the login name, the login program denies access to the system.

You can log failed command-line login attempts in the /var/adm/loginlog file. This is a useful tool if you want to determine if attempts are being made to break into a system.

By default, the loginlog file does not exist. To enable logging, you should create this file with read and write permissions for the root user only, and it should belong to the sys group.

```
# touch /var/adm/loginlog
# chown root:sys /var/adm/loginlog
# chmod 600 /var/adm/loginlog
```

All failed command-line login activity is written to this file automatically after five consecutive failed attempts.

The loginlog file contains one entry for each of the failed attempts. Each entry contains the user's login name, login device (TTY port), and time of the failed attempt.

If there are fewer than five consecutive failed attempts, no activity is logged to this file.

Switching Users on a System

As the system administrator, you should log in to a system as a regular user, and then switch to the root account only to perform administrative tasks.

You should avoid logging in directly as the root user. This precaution helps protect the system from unauthorized access, because it reduces the likelihood that the system will be left unattended with the root user logged in. Also, critical mistakes are less likely to occur if you perform routine work as a regular system user.

Introducing the su Command

Use the su command to switch to the superuser or another user without logging out and back in as that user.

The following is the command format for the su command:

```
su - username
```

If no user name is given, then the su command attempts to switch to the root user.

To use the su command, supply the appropriate password unless you are already the root user. The root user can run the su command without passwords.

If the password is correct, the su command creates a new shell process, as specified in the shell field of that user account's /etc/passwd file entry.

The su - (dash) option specifies a complete login by reading all of the user's shell initialization files. The - (dash) option changes your work environment to what would be expected if you had logged in directly as that specified user. It also changes the users home directory.

When you run the su command, the effective user ID (EUID) and the effective group ID (EGID) are changed to the new user to whom you have switched.

Access to files and directories is determined by the value of the EUID and EGID for the effective user, rather than by the UID and GID numbers of the original user who logged in to the system.

Using the whoami Command

The whoami command displays the name of the account, whose authorization you have switched to.

Note – The whoami command resides in the /usr/ucb directory.

For example, user1 is logged into the system under that login name. This user then runs the su command to become the root user and enters the root password. The whoami command displays the user's actual authorization for accessing directories and files, for example:

```
$ whoami
user1
$ pwd
/export/home/user1
$ su
password: EnterPassword
# whoami
root
# pwd
/export/home/user1
```

Using the who am i Command

To determine the login name of the original user, use the who command with the am i option.

To use the who am i command, at the shell prompt, type the su command and the login name of the user account to which you want to switch, and press Return. Type the password for the user account, and press Return.

For exam le, while logged in as user1, use the su command to switch to user2:

```
$ su user2
password:   EnterPassword
$ who am i
user1       pts/2        Dec 17 12:18      (129.147.4.12)
```

An alternative to the who am i command is the who -m command.

Switching to Another Regular User

To switch to another user and have that user's environment, use the su command as follows:

1. At the shell prompt, display your login name and path.

```
$ who am i
user1       pts/4          Feb  8 08:38
$ pwd
/export/home/user1
```

2. Enter the su command with the dash (-) option and the login name of the user to which you want to switch. Then, enter the password for the user.

```
$ su - user2
Password: EnterPassword
```

3. To determine the login name of the actual user, perform the whoami command, and press Return.

```
$ whoami
user2
```

4. To determine the current working directory, perform the pwd command. The location is the effective user's home directory.

```
$ pwd
/export/home/user2
```

5. To display the login name of the original user, perform the who am i command.

```
$ who am i
user1       pts/4          Feb  8 08:38
```

6. To return to the original user status and home directory, perform the exit command.

```
$ exit
$ pwd
/export/home/user1
```

Becoming the root User

In the default system configuration, direct root logins are restricted to the console. This means that you cannot remotely log in to a system as root. To remotely log in to a host as the root user, you must log in as a regular user and then run the su command to become the root user.

To become the root user, use the su command as follows:

1. Log in from the login window as a regular user, such as user1.
2. At the shell prompt in a terminal window, perform the su command. Enter the root password.

```
$ su -
Password: EnterPassword
```

3. To display the original login, perform the who am i command.

```
# who am i
user1        pts/4          Feb  8 08:48
```

4. To determine the login name of the user to which you have switched, perform the whoami command.

```
# whoami
root
```

5. To determine the current working directory, perform the pwd command.

```
# pwd
/
```

6. To exit the root session and return to the original user, perform the exit command.

```
# exit
$ pwd
/export/home/user1
$
```

Monitoring su Attempts

For security reasons, you must monitor who has been using the su command, especially those users who are trying to gain root access on the system. You can initiate the monitoring by setting two variables in the /etc/default/su file.

Note – There are many variables in the /etc/default/su file. This course presents only a small subset of the variables.

Contents of the /etc/default/su File

To display the contents of the /etc/default/su file, perform the command:

```
# cat /etc/default/su
#ident  "@(#)su.dfl     1.[illegible]     93/08/14 SMI"   /* SVr4.0 1.2   */

# SULOG determines the location of the file used to log all su attempts
SULOG=/var/adm/sulog

# CONSOLE determines whether attempts to su to root should be logged
# to the named device
#CONSOLE=/dev/console
(output edited for brevity)
SYSLOG=YES
```

In the preceding example, unsuccessful attempts to use the su command to access the root account are logged to the /var/adm/messages file. The following is an example entry from that file:

```
Dec 17 12:35:47 sys41 su: [ID 810491 auth.crit] 'su root' failed for
user3 on /dev/pts/2
```

The CONSOLE Variable in the /etc/default/su File

By default, the system ignores the CONSOLE variable in the /etc/default/su file because of the preceding comment (#) symbol. All attempts to use the su command are logged to the console, regardless of success or failure. Here is an example of output to the console:

```
Feb 2 09:50:09 host1 su: 'su root' failed for user1 on /dev/pts/4
Feb 2 09:50:33 host1 su: 'su user3' succeeded for user1 on /dev/pts/4
```

When the comment symbol is removed, the value of the CONSOLE variable is defined for the /dev/console file. Subsequently, an additional line of output for each successful attempt to use the su command to access the root account is logged to the console. Here is an example of logged su command activity:

```
Feb 2 11:20:07 host1 su: 'su root' succeeded for user1 on /dev/pts/4
SU 02/02 11:20 + pts/4 user1-root
```

The SULOG Variable in the /etc/default/su File

The SULOG variable in the /etc/default/su file specifies the name of the file in which all attempts to use the su command to switch to another user are logged. If the variable is undefined, the su command logging is turned off.

The /var/adm/sulog file is a record of all attempts by users on the system to execute the su command. Each time the su command is executed, an entry is added to the sulog file.

The entries in this file include the date and time the command was issued, whether it was successful (shown by the plus (+) symbol for success or the hyphen (-) symbol for failure), the device from which the command was issued, and, finally, the login and the effective identity.

The following is an example of entries from the /var/adm/sulog file:

```
# more /var/adm/sulog
SU 10/20 14:50 + console root-sys
SU 10/20 16:55 + pts/2 user1-root
SU 11/05 11:21 - pts/3 user1-root
```

Controlling System Access

The more access that is available over the network, the more beneficial it is for remote system users. However, unrestricted access and sharing of data and resources can create security problems.

A local host's remote security measures are generally based on an ability to validate, limit, or block operations from remote system users.

The /etc/default/login File

Note – There are many variables in the /etc/default/login file. This course, presents only a small subset of the variables.

The /etc/default/login file establishes default parameters for users when they log in to the system. The /etc/default/login file gives you the ability to protect the root account on a system. You can restrict root access to a specific device or to a console, or disallow root access altogether.

To display the contents of the /etc/default/login file, perform the command:

```
# cat /etc/default/login
(output edited for brevity)
# If CONSOLE is set, root can only login on that device.
# Comment this line out to allow remote login by root.
#
CONSOLE=/dev/console

# PASSREQ determines if login requires a password.
#
PASSREQ=YES
```

The CONSOLE Variable in the /etc/default/login File

You can set the CONSOLE variable in the /etc/default/login file to specify one of three possible conditions that restrict access to the root account:

- If the variable is defined as CONSOLE=/dev/console, the root user can log in only at the system console. Any attempt to log in as root from any other device generates the error message:

```
# rlogin host1
Not on system console
Connection closed.
```

- If the variable is not defined, such as #CONSOLE=/dev/console, the root user can log in to the system from any device across the network, through a modem, or using an attached terminal.
- If the variable does not have a value assigned to it (for example CONSOLE=) then the root user cannot log in from anywhere, not even the console. The only way to become the root user on the system is to log in as a regular user and then become root by using the su command.

Note – You can confine root logins to a particular port with the CONSOLE variable. For example, CONSOLE=/dev/term/a permits the root user to log in to the system only from a terminal that is connected to Serial Port A.

The PASSREQ Variable in the /etc/default/login File

When the PASSREQ variable in the /etc/default/login file is set to the default value of YES, then all users who had not been assigned passwords when their accounts were created are required to enter a new password as they log in for the first time. If this variable is set to NO, then null passwords are permitted. This variable does not apply to the root user.

File Transfer Protocol (FTP) Access

The Solaris OE provides an American Standard Code for Information Interchange (ASCII) file named `/etc/ftpd/ftpusers`. The `/etc/ftpd/ftpusers` file lists the names of users who are prohibited from connecting to the system through the FTP protocol.

Each line entry in this file contains a login name for a restricted user, for example:

```
username
```

The FTP server daemon `in.ftpd` reads the `/etc/ftpd/ftpusers` file when an FTP session is invoked. If the login name of the user matches one of the listed entries, it rejects the login session and sends the `Login failed` error message.

By default, the `/etc/ftpd/ftpusers` file lists these system account entries:

```
root
daemon
bin
sys
adm
lp
uucp
nuucp
smmsp
listen
nobody
noaccess
nobody4
```

As with any login name that you can add, these entries must match the user account names located in the `/etc/passwd` file.

The `root` entry is included in the `ftpusers` file as a security measure. The default security policy is to disallow remote logins for the `root` user. The policy is also followed for the default value set as the `CONSOLE` entry in the `/etc/default/login` file.

The /etc/hosts.equiv and $HOME/.rhosts Files

Typically, when a remote user requests login access to a local host, the first file read by the local host is its /etc/passwd file. An entry for that particular user in this file enables that user to log in to the local host from a remote system. If a password is associated with that account, then the remote user is required to supply this password at log in to gain system access.

If there is no entry in the local host's /etc/passwd file for the remote user, access is denied.

The /etc/hosts.equiv and $HOME/.rhosts files bypass this standard password-based authentication to determine if a remote user is allowed to access the local host, with the identity of a local user.

These files provide a remote authentication procedure to make that determination.

This procedure first checks the /etc/hosts.equiv file and then checks the $HOME/.rhosts file in the home directory of the local user who is requesting access. The information contained in these two files (if they exist) determines if remote access is granted or denied.

The information in the /etc/hosts.equiv file applies to the entire system, while individual users can maintain their own $HOME/.rhosts files in their home directories.

Figure 11-2 shows the flow of remote access authentication.

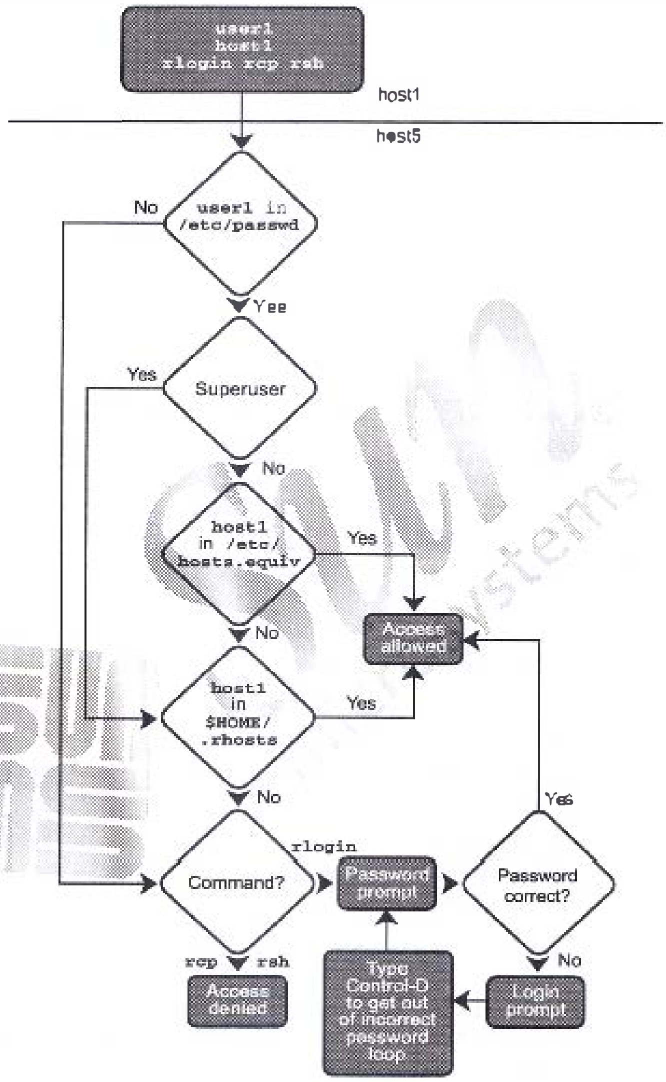

Figure 11-2 Remote Access Authentication

Entries in the /etc/hosts.equiv and $HOME/.rhosts Files

While the /etc/hosts.equiv and $HOME/.rhosts files have the same format, the same entries in each file have different effects.

Both files are formatted as a list of one-line entries, which can contain the following types of entries:

hostname
hostname username
+

The host names in the /etc/hosts.equiv and $HOME/.rhosts files must be the official name of the host, not one of its alias names.

Note – When logging in to a number of different systems, you can run the uname -n command to determine on which system you are currently logged in.

The /etc/hosts.equiv File Rules

For regular users, the /etc/hosts.equiv file identifies remote hosts and remote users who are considered to be *trusted*.

Note – The /etc/hosts.equiv file is not checked at all if the remote user requesting local access is the root user.

If the local host's /etc/hosts.equiv file contains the host name of a remote host, then all regular users of that remote host are trusted and do not need to supply a password to log in to the local host. This is provided so that each remote user is known to the local host by having an entry in the local /etc/passwd file; otherwise, access is denied.

This functionality is particularly useful for sites where regular users commonly have accounts on many different systems, eliminating the security risk of sending ASCII passwords over the network.

The /etc/hosts.equiv file does not exist by default. It must be created if trusted remote user access is required on the local host.

The $HOME/.rhosts File Rules

While the /etc/hosts.equiv file applies system-wide access for non-root users, the .rhosts file applies to a specific user.

All users, including the root user, can create and maintain their own .rhosts files in their home directories.

For example, if you run an rlogin process from a remote host to gain root access to a local host, the /.rhosts file is checked in the root home directory on the local host.

If the remote host name is listed in this file, it is a trusted host, and, in this case, root access is granted on the local host. The CONSOLE variable in the /etc/default/login file must be commented out for remote root logins.

The $HOME/.rhosts file does not exist by default. You must create it in the user's home directory.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: User Access (Level 1)

In this exercise, you complete the following tasks:

- Log failed login attempts
- Use the commands finger, last, rusers, su, and whoami
- Examine the sulog file
- Change the /etc/default/login file to allow root logins from any terminal
- Change the /etc/ftpd/ftpusers file to allow FTP access as the root user
- Create a /.rhosts file to allow root access from another system

Preparation

This lab requires two systems. Each system lists the other in its /etc/inet/hosts file. The lab also requires two specific users, user9 and user3, on both systems. Both users should use the password 123pass. Refer to the lecture notes as necessary to perform the steps listed.

Tasks

Complete the following tasks:

- Create a log file to record failed login attempts. Use the command-line login to make five failed login attempts. List the contents of the log file. Use commands to display information for user3 on both your system and your partner's system.

  (Steps 1–7 in the Level 2 lab)
- Identify when the first root login session on your system occurred and how long the session lasted. Identify when your system last booted. List the users logged in on all systems on your network and on just your partner's system.

  (Steps 8–11 in the Level 2 lab)

- Change your user identity from the root user to user9, both with and without the - (dash) option. Record the differences. List effective and real user identity during your su sessions. Locate the su log and identify which user initiated your su attempts.

  (Steps 12–18 in the Level 2 lab)

- As the root user, attempt to log into your partner's system. Record error messages. Change the CONSOLE variable on your partner's system to allow root logins from any terminal. Attempt to access your partner's system again.

  (Steps 19–21 in the Level 2 lab)

- As the root user, attempt to use the ftp command to access your partner's system. Change the ftp permissions file to allow root access to your partner's system.

  (Step 22 in the Level 2 lab)

- As the root user, attempt to use the rlogin command to access your partner's system. Ask your partner to create a /.rhosts file that lists your system name. Attempt to use the rlogin command to access your partner's system again.

  (Step 23 in the Level 2 lab)

Exercise: User Access (Level 2)

In this exercise, you complete the following tasks:

- Log failed login attempts
- Use the commands `finger`, `last`, `rusers`, `su`, and `whoami`
- Examine the `sulog` file
- Change the `/etc/default/login` file to allow `root` logins from any terminal
- Change the `/etc/ftpd/ftpusers` file to allow FTP access as the `root` user
- Create a `/.rhosts` file to allow root access from another system

Preparation

This lab requires two systems. Each system lists the other in its `/etc/inet/hosts` files. It also requires two specific users, `user9` and `user3`, on both systems. Both users should use the password `123pass`. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

In this exercise, you accomplish the following:

- Create the file `/var/adm/loginlog`. Use the command-line login to make five failed login attempts. List the contents of the `/var/adm/loginlog` file. Use the `finger` command to display information for `user9` on both your system and your partner's system.
- Use the `last` command to identify when the first `root` login session on your system occurred and how long the session lasted. Use the `last` command to learn when your system last booted. Use the `rusers` command to list the users logged in on all systems on your network and on just your partner's system.

- Use the su command to change your user identity from the root user to user9, both with and without the - (dash) option. Record the differences. Use the whoami and who am i commands to list your effective and real user identity during your su sessions. Locate the su log declared in the /etc/default/su file, and identify which user initiated your su attempts.
- As the root user, attempt a session to your partner's system by using the telnet command. Record error messages. Change the CONSOLE variable on your partner's system to allow root logins from any terminal. Attempt the telnet session again.
- As the root user, attempt to use the ftp command to access your partner's system. Change the /etc/ftpd/ftpusers file to allow root access to your partner's system.
- As the root user, attempt to use the rlogin command to access your partner's system. Ask your partner to create a /.rhosts file that lists your system name. Attempt to use the rlogin command to access your partner's system again.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to /var/adm.
2. Use the touch command to create a file called loginlog. (Ensure permissions are set to read and write for the root user only.) If necessary, set the group ownership to sys.
3. Log out. From the CDE Options menu, select the Command Line Login option. When the CDE login screen clears, press Return to obtain the command-line login prompt.
4. Enter root after the login prompt, but supply an incorrect password. Do this five times. After the fifth attempt, the CDE login screen appears again. Log in as root, and open a terminal window.
5. Examine the /var/adm/loginlog file. What does it contain?
6. Use the finger command to display information for the user called user9. What is the difference in the output between the finger -m command and the finger command with no option?

7. Use the finger command to display information for the same user on your partner's system. (You will need to reference your partner's system on the command line.) Try this with and without the -m option. Does the -m option change the output that the finger command displays?
8. Use the last command to display login and system reboot activity. When did the first root login occur, and how long did that session last?
9. Use the last command to display only system boot activity. When did the system last reboot?
10. Use the rusers command to list information about the users on all systems on your network segment.
11. Use the rusers command to list information for users on your partner's system. When, and on what terminal, did the first user listed log in?
12. Switch your user identity to that of user9. Do not use the – (dash) option.
13. Display some of the variables that define your environment.
14. Exit the su session and try to switch your user identity again, this time using the – (dash) option.

    Are the values reported now correct for the user root or for user9?
15. Use the whoami and who am i commands to list your effective and real user identity.

    What do these commands report?
16. Use the su command to change your user identity from user9 to user3, and use the whoami and who am i commands again.

    What do these commands report?

    Exit both su sessions when you are finished.
17. Change the directory to /etc/default. Examine the /etc/default/su file, and record the value of the SULOG variable.
18. Display the file named by the SULOG variable, and identify the entry that relates to your last su command. Is user9 or the root user identified as the user who became user3?
19. As the user root, attempt to log in to your partner's system by using the telnet command. Was your attempt successful? What message appears?

20. On your partner's system, edit the `/etc/default/login` file, and change the line that reads:

```
CONSOLE=/dev/console
```

so that it reads:

```
#CONSOLE=/dev/console
```

21. As the `root` user, again attempt to log in to your partner's system by using the `telnet` command. If your login attempt is successful, exit the `telnet` session. If not, check the change you made in Step 20, and try again.
22. As the `root` user, attempt to use the `ftp` command to access your partner's system. Were you successful? Ask your partner to edit the `/etc/ftpd/ftpusers` file and comment out the `root` entry. Attempt to use the `ftp` command to access your partner's system again. List some files in the `/tmp` directory from the `ftp>` prompt.
23. As the `root` user, attempt to use the `rlogin` command to access your partner's system. Were you successful? Ask your partner to create a `/.rhosts` file and enter the name of your system on a line by itself. Attempt to use the `rlogin` command to access your partner's system again.

Exercise: User Access (Level 3)

In this exercise, you complete the following tasks:

- Log failed login attempts
- Use the commands `finger`, `last`, `rusers`, `su`, and `whoami`
- Examine the `sulog` file
- Change the `/etc/default/login` file to allow `root` logins from any terminal
- Change the `/etc/ftpd/ftpusers` file to allow FTP access as the `root` user
- Create a `/.rhosts` file to allow root access from another system

Preparation

This lab requires two systems that list each other in their `/etc/inet/hosts` files. It also requires two specific users, `user9` and `user3`, on both systems. Both users should use the password `123pass`. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

In this exercise, you accomplish the following:

- Create the file `/var/adm/loginlog`. Use the command-line login to make five failed login attempts. List the contents of the `/var/adm/loginlog` file. Use the `finger` command to display information for `user9` on both your system and your partner's system.
- Use the `last` command to identify when the first `root` login session on your system occurred and how long the session lasted. Use the `last` command to learn when your system last booted. Use the `rusers` command to list the users logged in on all systems on your network and on just your partner's system.

- Use the `su` command to change your user identity from the `root` user to `user9`, both with and without the – (dash) option. Record the differences. Use the `whoami` and `who am i` commands to list your effective and real user identity during your `su` sessions. Locate the `su` log declared in the `/etc/default/su` file, and identify which user initiated your `su` attempts.
- As the `root` user, attempt a session to your partner's system by using the `telnet` command. Record error messages. Change the `CONSOLE` variable on your partner's system to allow `root` logins from any terminal. Attempt the `telnet` session again.
- As the `root` user, attempt to use the `ftp` command to access your partner's system. Change the `/etc/ftpd/ftpusers` file to allow `root` access to your partner's system.
- As the `root` user, attempt to use the `rlogin` command to access your partner's system. Ask your partner to create a `/.rhosts` file that lists your system name. Attempt to use the `rlogin` command to access your partner's system again.

Tasks and Solutions

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Change the directory to `/var/adm`.

```
# cd /var/adm
```

2. Use the `touch` command to create a file called `loginlog`. (Ensure permissions are set to read and write for the `root` user only.) If necessary, set the group ownership to `sys`.

```
# touch loginlog
# chmod 600 loginlog
# chgrp sys loginlog
```

3. Log out. From the CDE Options menu, select the Command Line Login option. When the CDE login screen clears, press Return to obtain the command-line login prompt.
4. Enter `root` after the login prompt, but supply an incorrect password. Do this five times. After the fifth attempt, the CDE login screen appears again. Log in as `root`, and open a terminal window.

5. Examine the /var/adm/loginlog file. What does it contain?

   This file should contain a list of failed login attempts which appear similar to the following:

   ```
   login:/dev/pts/2:Tue Dec 18 13:29:22 2001
   ```

6. Use the `finger` command to display information for the user called `user9`. What is the difference in output between the `finger -m` command and the `finger` command with no option?

```
# finger user9
# finger -m user9
```

   The `finger` command with no option lists all user accounts that have the string `user` in their names and comment fields. The `finger -m` command lists only the entry for the user named `user9`.

7. Use the `finger` command to display information for the same user on your partner's system. (You will need to reference your partner's system on the command line.) Try this with and without the `-m` option. Does the `-m` option change the output that the `finger` command displays?

```
# finger user9@hostname
# finger -m user9@hostname
```

   No.

8. Use the `last` command to display login and system reboot activity. When did the first `root` login occur, and how long did that session last?

```
# last
```

   This information depends on the activity on your particular system.

9. Use the `last` command to display only system boot activity. When did the system last reboot?

```
# last reboot
```

   This information depends on the activity on your particular system.

10. Use the `rusers` command to list information about the users on all systems on your network segment.

```
# rusers -l
```

11. Use the `rusers` command to list information about the users on your partner's system. When, and on what terminal, did the first user listed log in?

```
# rusers -l hostname
```

   This information depends on the activity on your particular system.

12. Switch your user identity to that of user9. Do not use the - (dash) option.

```
# su user9
$
```

13. Display some of the variables that define your environment.

```
$ echo $LOGNAME
$ echo $HOME
```

Are the values reported correct for the user root or for user9?

root

14. Exit the su session and try to switch your user identity again, this time using the - (dash) option.

```
$ exit
# su - user9
$ echo $LOGNAME
$ echo $HOME
```

Are the values reported now correct for the user root or for user9?

user9

15. Use the whoami and who am i commands to list your effective and real user identity.

```
$ /usr/ucb/whoami
$ who am i
```

What do these commands report?

The /usr/ucb/whoami command displays the login name matching your effective UID, user9. The who am i command displays the login name matching your real UID, root.

16. Use the su command to change your user identity from user9 to user3, and use the whoami and who am i commands again.

```
$ su user3
Password: 123pass
$
```

What do these commands report?

```
$ /usr/ucb/whoami
```

user3

```
$ who am i
```

root

Exit both su sessions when you are finished.

```
$ exit
$ exit
#
```

17. Change the directory to /etc/default. Examine the /etc/default/su file, and record the value of the SULOG variable.

```
# cd /etc/default
# more su
```

/var/adm/sulog

18. Display the file named by the SULOG variable, and identify the entry that relates to your last su command. Is user9 or the root user identified as the user who became user3?

```
# cat /var/adm/sulog
```

root

19. As the root user, attempt to log in to your partner's system by using the telnet command. Was your attempt successful? What message appears?

```
# telnet hostname
(telnet connection messages)

SunOS 5.9

login: root
Password: cangetin
```

The login attempt should not succeed. It fails and the system sends the messages:

```
Not on system console
Connection closed by foreign host.
```

20. On your partner's system, edit the /etc/default/login file, and change the line that reads:

```
CONSOLE=/dev/console
```

so that it reads:

```
#CONSOLE=/dev/console
```

21. As the root user, again attempt to log in to your partner's system by using the telnet command. If your login attempt is successful, exit the telnet session. If not, check the change you made in Step 20, and try again.

```
# telnet host
(telnet connection messages)
SunOS 5.9

login: root
Password: cangetin
Last login: Fri Feb  8 08:38:17 from sys41
Sun Microsystems Inc.   SunOS 5.9       s81_54  May 2002

# exit
Connection closed by foreign host.
#
```

22. As the root user, attempt to use the ftp command to access your partner's system. Were you successful?

No, you should receive the message: Login incorrect. Login failed.

Ask your partner to edit the /etc/ftpd/ftpusers file and comment out the root entry. Attempt to use the ftp command to access your partner's system again. List some files in the /tmp directory from the ftp> prompt.

You should see files such as:

dtdbcache_:0

sdtvolcheck402

speckeysd.lock

23. As the root user, attempt to use the rlogin command to access your partner's system. Were you successful?

You should not be able to use the rlogin *command to directly access your partner's system. You should be prompted for a password.*

Ask your partner to create a /.rhosts file and enter the name of your system on a line by itself. Attempt to use the rlogin command to access your partner's system again.

You should be able to use the rlogin *command to log directly in to your partner's system now.*

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Restricting Access to Data in Files

After you have established login restrictions, the next task is to control access to the data on the systems. Of course, some users need to be allowed to read various files; other users need permission to change and delete files, and there are some files that no regular user should be able to access.

Users who need to share files should be in the same group in the /etc/group file.

Note – In general, you use file access permissions to determine which users or groups have permission to read, modify, or delete files.

Determining a User's Group Membership

The groups command displays group memberships for the user.

The command format for the groups command is:

```
groups [username]
```

For example, to see which groups you are a member of, perform the command:

```
# groups
other root bin sys adm uucp mail tty lp nuucp daemon
```

To list the groups to which a specific user is a member, use the groups command with the user's name, such as user5, as an argument.

```
# groups user5
staff class sysadmin
```

Identifying a User Account

You use the id command to further identify users by listing their UID number, user name, GID number, and group name. This information is useful when you are troubleshooting file access problems for users.

The id command also returns the EUID number and name, and the EGID number and login name. For example, if you logged in as user1 and then used the su command to become user4, the id command reports the information for the user4 account.

The command format for the id command is:

```
id options username
```

To view your effective user account, perform the command:

```
$ id
uid=101(user1) gid=300(class)
```

To view account information for a specific user, use a user login name with the id command:

```
$ id user1
uid=101(user1) gid=300(class)
```

To view information about the secondary groups of a user, use the -a option and a user login name, such as user1:

```
$ id -a user1
uid=101(user1) gid=300(class) groups=14(sysadmin)
```

Changing File and Directory Ownership

You might need to use the chown command to change the original owner of a file or directory to another user account on the system. By default, only the root user can change the ownership of a file or directory.

Note – Regular users can be given permission to use the chown command to change the ownership of files and directories owned by them. Edit the /etc/system file, and add the parameter: set rstchown=0 (zero). You need to reboot the system for the changes to take effect.

The command format for the chown command is:

```
chown option(s) user_name filename(s)
```

or

```
chown option(s) UID filename(s)
```

Note – The user must exist in the /etc/passwd file.

In this example, a user named user1 created a file called file7.

```
# cd /export/home/user1
# ls -l file7
-rw-r--r--   1  user1     staff          672 Jun 1 15:11     file7
#
```

You can use the chown command to give ownership of this file to a new user named user2. You use the ls command to verify the new ownership.

```
# chown user2 file7
# ls -l file7
-rw-r--r--   1  user2     staff          672 Jun 1 15:12     file7
#
```

After this sequence of commands, the file is owned by user2. This file is still in the home directory of user1. The two users need to determine if the file should be moved to a new directory location.

The ownerships of subdirectories can be changed in the same manner as files, as shown in the following examples:

In this example, user1 owns a directory called dir4.

```
$ ls -lR dir4
dir4:
total 0
-rw-r--r--   1 user1    staff          0 Mar 19 16:06 file1
-rw-r--r--   1 user1    staff          0 Mar 19 16:06 file2
-rw-r--r--   1 user1    staff          0 Mar 19 16:06 file3
$
```

You would use the chown command with the -R option to give ownership of this directory and all of its contents (files and subdirectories) to user2.

```
$ chown -R user2 dir4
$ ls -lR dir4
dir4:
total 0
-rw-r--r--   1 user2    staff           0 Mar 19 16:06 file1
-rw-r--r--   1 user2    staff           0 Mar 19 16:06 file2
-rw-r--r--   1 user2    staff           0 Mar 19 16:06 file3
$
```

The -R option makes the chown command recursive. It descends through the directory and any subdirectories, setting the ownership UID number as it moves through the directory hierarchy.

The chown command can also change both the individual and group ownership of a file or subdirectory simultaneously.

```
$ chown user3:class file2
```

Additionally, you can use the -R option to descend a directory hierarchy recursively, changing individual and group ownership of the directory and its contents simultaneously. The following example demonstrates this kind of change to the dir1 directory.

```
$ chown -R user3:class dir1
$ ls -lR dir1
dir1:
total 0
-rw-r--r--   1 user3    class           0 Mar 19 16:18 file1
-rw-r--r--   1 user3    class           0 Mar 19 16:18 file2
```

Changing File and Directory Group Membership

The chgrp command can be used by the root user or the file's owner to change the group ownership of files and directories to another group on the system. However, the file owner must also belong to the new group.

Note – Regular users can be given permission to use the chgrp command to change a file's or directory's group ownership to groups of which the user is not a member. Edit the /etc/system file, and add a parameter: set rstchown=0 (zero). You must reboot the system for the changes to take effect.

The command format for the chgrp command is:

```
chgrp groupname filename(s)
```

or

```
chgrp GID filename(s)
```

Note – The *groupname* must exist in the /etc/group file.

For example, the file4 file currently is a member of a group named staff.

```
# ls -l file4
-rw-rw-r--   1  user1    staff          874 Jun 1 15:08    file4
#
```

You would use the chgrp command to give this file to a new group named class and use the ls command to verify the new group ownership.

```
# chgrp class file4
# ls -l file4
-rw-rw-r--   1  user1    class          874 Jun 1 15:09    file4
#
```

When you are finished, all users who are members of the group called class have read and write access to this file.

Using File Permissions

Three types of special permissions are available for executable files and directories. These are:

- The setuid permission
- The setgid permission
- The Sticky Bit permission

The setuid Permission on Executable Files

When the set-user identification (setuid) permission is set on an executable file, a user or process that runs this executable file is granted access based on the owner of the file (usually the root user), instead of on who started the executable.

This setting allows a user to access files and directories that are typically accessible only by the owner of the executable. Note that many executable programs must be run by the root user, or by sys or bin to work properly.

Use the ls command to check the setuid permission.

```
# ls -l /usr/bin/su
-r-sr-xr-x   1 root     sys      22292 Jan 15 17:49 /usr/bin/su
```

The setuid permission displays as an "s" in the owner's execute field.

Note – If a capital "S" appears in the owner's execute field, it indicates that the setuid bit is on, and the execute bit "x" for the owner of the file is off or denied.

The root user and the owner can set the setuid permissions on an executable file by using the chmod command and the octal value 4###.

For example:

```
# chmod 4555 executable_file
```

Except for those setuid executable files that exist by default in the Solaris OE, you should disallow the use of setuid programs or at least restrict their use.

To search for files with setuid permissions and to display their full path names, perform the command:

```
# find / -perm -4000
```

The setgid Permission on Executable Files

The set-group identification (setgid) permission is similar to the setuid permission, except that when the process runs, it runs as if it were a member of the same group in which the file is a member. Also, access is granted based on the permissions assigned to that group.

For example, the write program has a setgid permission that allows users to send messages to other users' terminals.

Use the ls command to check the setgid permission.

```
# ls -l /usr/bin/write
-r-xr-sr-x   1 root     tty        11484 Jan 15 17:55 /usr/bin/write
```

The setgid permission displays as an "s" in the group's execute field.

Note – If a lowercase letter "l" appears in the group's execute field, it indicates that the setgid bit is on, and the execute bit for the group is off or denied. This indicates that mandatory file and record locking occurs during file access for those programs that are written to request locking.

The root user and the owner can set setgid permissions on an executable file by using the chmod command and the octal value 2###. Here is the command-line format:

```
# chmod 2555 executable_file
```

The setgid Permission on Directories

The setgid permission is a useful feature for creating shared directories.

When a setgid permission is applied to a directory, files created in the directory belong to the group of which the directory is a member.

For example, if a user has write permission in the directory and creates a file there, that file is a member of the same group as the directory and not the user's group.

To create a shared directory, you must set the setgid bit using symbolic mode. Here is the format for that mode:

```
# chmod g+s shared_directory
```

To search for files with setgid permissions and display their full path names, perform the command:

```
# find / -perm -2000
```

Sticky Bit Permission on Public Directories

The Sticky Bit is a special permission that protects the files within a publicly writable directory.

If the directory permissions have the Sticky Bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by the root user. This prevents a user from deleting other users' files from publicly writable directories.

Use the ls command to determine if a directory has the Sticky Bit permission set.

```
# ls -ld /tmp
drwxrwxrwt   6   root   sys      719 May 31 03:30    /tmp
```

The Sticky Bit displays as the letter "t" in the execute field for other.

Note – If a capital "T" appears in the execute field for other, it indicates that the Sticky Bit is on; however, the execute bit is off or denied.

The root user and the owner can set the Sticky Bit permission on directories by using the chmod command and the octal value 1###. Here is the command-line format:

```
# chmod 1777 public_directory
```

To search for directories that have Sticky Bit permissions and display their full path names, execute the following command:

```
# find / -type d -perm -1000
```

Note – For more detailed information on the Sticky Bit, execute the man sticky command.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Restricting Access to Data on Systems (Level 1)

In this exercise, you complete the following tasks:

- Practice using commands related to user identity and file ownership
- Assign a user to the sysadmin group
- Assign special file permissions to files

Preparation

Refer to lecture notes as necessary to perform the steps listed.

Tasks

Complete the following tasks:

- Using the commands presented in the lecture, identify the groups of which root is a member. Compare the output from these commands. Add a user account called user11 with the useradd command. Verify the list of groups of which user11 is a member. Use the Solaris Management Console to create a new user account called user12. Add user11 to the sysadmin group.

  (Steps 1–7 in the Level 2 lab)
- Log in as user11 and create a new file called file1. Attempt to change its user ownership. Record error messages. Change the group ownership of file1 to sysadmin. Switch the user identity to the root user, and change ownership of file1 to user12.

  (Steps 8–11 in the Level 2 lab)
- As user11, create a new file called file2. Set setuid and setgid permissions on file2. Remove all execute permissions from file2. Record the permissions listed as you change them.

  (Steps 12–15 in the Level 2 lab)

- Record the permissions associated with the `/tmp` directory. As `user11`, create a new file called `test1` in the `/tmp` directory. As `user12`, attempt to remove this file. Record the result. As `user11`, create a new directory called `dir1` in `/export/home/user11`. Set permissions for the `dir1` directory to 777. Create a file called `test2` in the `dir1` directory. As `user12` attempt to remove this file. Record the result. Log in again as the `root` user.

  (Steps 16–21 in the Level 2 lab)

Exercise: Restricting Access to Data on Systems (Level 2)

In this exercise, you complete the following tasks:

- Practice using commands related to user identity and file ownership
- Assign a user to the sysadmin group
- Assign special file permissions to files

Preparation

Refer to lecture notes as necessary to perform the steps listed.

Task Summary

In this exercise, you accomplish the following:

- Using the commands groups, id, and id -a, identify the groups of which the root user is a member. Compare the output from these commands. Add a user account called user11 with the useradd command. Verify the list of groups of which user11 is a member. Use the Solaris Management Console to create a new user account called user12. Add user11 to the sysadmin group.
- Log in as user11 and create a new file called file1. Attempt to change its user ownership. Record error messages. Change the group ownership of file1 to sysadmin. Switch your user identity to the root user, and change ownership of file1 to user12.
- As user11, create a new file called file2. Use the chmod command to set setuid and setgid permissions on file2. Use the chmod command to remove all execute permissions from file2. Record the permissions listed as you change them.
- Record the permissions associated with the /tmp directory. As user11, create a new file called test1 in the /tmp directory. As user12, attempt to remove this file. Record the result. As user11, create a new directory called dir1 in /export/home/user11. Set permissions for the dir1 directory to 777. Create a file called test2 in the dir1 directory. As user12 attempt to remove this file. Record the result. Log in again as the root user.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Use the `groups` command to display the groups of which `root` is a member. Record the list that the `groups` command displays.
2. Use the `id` command both without and then with the `-a` option.

   Does the `id` command report the primary or a secondary group for the `root` user?

   Compare the `id -a` command output with that from the `groups` command in Step 1. What additional information does the `id -a` command provide?
3. Use the `useradd` command to create a new user account called `user11` with the following characteristics:

User Name:	`user11`
User ID:	`1011`
Primary Group:	`10`
Login Shell:	Korn
Home Directory:	`/export/home/user11`
Comment:	`SA239 User`
Password:	`123pass`

4. List the groups of which `user11` is a member.
5. Open a terminal window, and launch the Solaris Management Console.

6. Open the User Accounts tool. Select Add User from the Action menu. Then select From Template. Create a user account from the following information. Exit the Solaris Management Console when you are finished.

User Name:	user12
User ID:	1012
Password:	123pass

7. From a terminal window, use the usermod command to add user11 to group 14. Verify that the change took place. Log out.
8. Log in as user11. Open a terminal window, and use the touch command to create a file called file1. Verify that user11 and the group staff own file1.
9. Attempt to change the owner of file1 from user11 to user12. What error message displays?
10. Attempt to change the group ownership of file1 from staff to sysadmin. Verify the change. Did it work?
11. Switch your user identity to the root user, and change the directory to /export/home/user11. Change the owner of file1 from user11 to user12. Verify the change. Did it work? Exit your su session when you are finished.
12. In the home directory for user11, use the touch command to create a file called file2. Display and record the permissions associated with file2.
13. Use the chmod command to add setuid and execute permissions to file2. Display and record the permissions associated with file2. What changed?
14. Use the chmod command to add setuid and setgid permissions to file2. Display and record the permissions associated with file2. What changed?
15. Use the chmod command with octal arguments to remove all execute permissions from file2. Display and record the permissions associated with file2. What changed?
16. Change the directory to / (root), and list the permissions associated with the /tmp directory. Is the Sticky Bit set on /tmp? Do all users have write permission in the /tmp directory?

17. Change the directory to `/tmp`. Create a file called `test1` in the `/tmp` directory. Verify that `user11` and the group `staff` own `test1` and that 644 (`rw-r--r--`) permissions apply. Do they?
18. Switch your user identity to `user12`. In the `/tmp` directory, attempt to remove the `test1` file. What messages appear? Exit your `su` session when you are finished.
19. In the home directory for `user11`, create a directory called `dir1`. Change permissions for the `dir1` directory to 777. Create a file called `test2` below the `dir1` directory.
20. Switch your user identity to `user12`. Attempt to remove the file `test2` from the `dir1` directory. Verify that the `test2` file no longer exists. Exit your `su` session when you are finished.
21. Log out, and log in again as the `root` user.

Exercise: Restricting Access to Data on Systems (Level 3)

In this exercise, you complete the following tasks:

- Practice using commands related to user identity and file ownership
- Assign a user to the sysadmin group
- Assign special file permissions to files

Preparation

Refer to lecture notes as necessary to perform the steps listed.

Task Summary

In this exercise, you accomplish the following:

- Using the commands groups, id, and id -a, identify the groups of which the root user is a member. Compare the output from these commands. Add a user account called user11 by using the useradd command. Verify the list of groups of which user11 is a member. Use the Solaris Management Console to create a new user account called user12. Add user11 to the sysadmin group.
- Log in as user11 and create a new file called file1. Attempt to change its user ownership. Record error messages. Change the group ownership of file1 to sysadmin. Switch your user identity to the root user, and change ownership of file1 to user12.
- As user11, create a new file called file2. Use the chmod command to set setuid and setgid permissions on file2. Use the chmod command to remove all execute permissions from file2. Record the permissions listed as you change them.
- Record the permissions associated with the /tmp directory. As user11, create a new file called test1 in the /tmp directory. As user12, attempt to remove this file. Record the result. As user11, create a new directory called dir1 in /export/home/user11. Set permissions for the dir1 directory to 777. Create a file called test2 in the dir1 directory. As user12 attempt to remove this file. Record the result. Log in again as the root user.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Use the groups command to display the groups of which root is a member. Record the list that the groups command displays.

```
# groups
```

```
other root bin sys adm uucp mail tty lp nuucp
daemon
```

2. Use the id command both without and then with the -a option.

```
# id
```

Does the id command report the primary or a secondary group for the root user?

The id command reports the primary group.

```
# id -a
```

Compare the id -a command output with that from the groups command in Step 1. What additional information does the id -a command provide?

The id -a command reports group ID numbers in addition to group names for all groups.

3. Use the useradd command to create a new user called user11 with the following characteristics:

User Name:	user11
User ID:	1011
Primary Group:	10
Login Shell:	Korn
Home Directory:	/export/home/user11
Comment:	SA239 User
Password:	123pass

```
# useradd -u 1011 -g 10 -d /export/home/user11 -m -s /bin/ksh -c "SA239
User" user11
64 blocks
# passwd user11
New password: 123pass
```

```
Re-enter new password: 123pass
passwd (SYSTEM): passwd successfully changed for user11
#
```

4. List the groups of which user11 is a member.

```
# id -a user11
```

staff

5. Open a terminal window, and run the Solaris Management Console.

```
# smc &
```

6. Open the User Accounts tool. Select Add User from the Action menu. Then select With Template. Create a user account from the following information. Exit the Solaris Management Console when you are finished.

User Name:	user12
User ID:	1012
Password:	123pass

7. From a terminal window, use the usermod command to add user11 to group 14. Verify that the change took place. Log out.

```
# usermod -G 14 user11
# id -a user11
```

8. Log in as user11. Open a terminal window, and use the touch command to create a file called file1. Verify that user11 and the group staff own file1.

```
$ touch file1
$ ls -l file1
```

9. Attempt to change the owner of file1 from user11 to user12. What error message appears?

```
$ chown user12 file1
chown: file1: Not owner
```

10. Attempt to change the group ownership of file1 from staff to sysadmin. Verify the change. Did it work?

```
$ chgrp sysadmin file1
$ ls -l file1
```

Yes.

11. Switch your user identity to the root user, and change the directory to /export/home/user11. Change the owner of file1 from user11 to user12. Verify the change. Did it work? Exit your su session when you are finished.

```
$ su -
Password: cangetin
# pwd
/
# cd /export/home/user11
# chown user12 file1
# ls -l
-rw-r--r--   1 user12 sysadmin 0 Apr 17  2002 file1
# exit
$
```

Yes.

12. In the home directory for user11, use the touch command to create a file called file2. Display and record the permissions associated with file2.

```
$ touch file2
$ ls -l file2
```

The permissions for file2 *should read* -rw-r--r.

13. Use the chmod command to add setuid and execute permissions to file2. Display and record the permissions associated with file2. What changed?

```
$ chmod 4555 file2
$ ls -l file2
```

The permissions for file2 *should read* -r-sr-xr-x.

14. Use the chmod command to add setuid and setgid permissions to file2. Display and record the permissions associated with file2. What changed?

```
$ chmod 6555 file2
$ ls -l file2
```

The permissions for file2 *should read* -r-sr-sr-x.

15. Use the chmod command with octal arguments to remove all execute permissions from file2. Display and record the permissions associated with file2. What changed?

```
$ chmod 6444 file2
$ ls -l file2
```

The permissions for file2 *should read* -r-Sr-lr--.

16. Change the directory to / (root), and list the permissions associated with the /tmp directory. Is the Sticky Bit set on the /tmp directory? Do all users have write permission in /tmp?

```
$ cd /
$ ls -ld tmp
```

Yes to both.

17. Change the directory to /tmp. Create a file called test1 in the /tmp directory. Verify that user11 and the group staff own test1 and that 644 (rw-r--r--) permissions apply. Do they?

```
$ cd tmp
$ touch test1
$ ls -l test1
```

Yes.

18. Switch your user identity to user12. In the /tmp directory, attempt to remove the test1 file. What messages appear? Exit your su session when you are finished.

```
$ su user12
Password: 123pass
$ rm test1
rm: test1: override protection 644 (yes/no)? y
rm: test1 not removed: Permission denied
$ exit
$
```

19. In the home directory for user11, create a directory called dir1. Change permissions for the dir1 directory to 777. Create a file called test2 below the dir1 directory.

```
$ cd
$ mkdir dir1
$ chmod 777 dir1
$ touch dir1/test2
```

20. Switch your user identity to user12. Attempt to remove the file test2 from the dir1 directory. Verify that the test2 file no longer exists. Exit your su session when you are finished.

```
$ su user12
Password: 123pass
$ rm dir1/test2
$ ls -l dir1
$ exit
$
```

21. Log out, and log in again as the root user.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 12

Configuring Printer Services

Objectives

Upon completion of this module, you should be able to:

- Identify network printing fundamentals
- Configure printer services
- Administer printer services
- Start and stop the line printer (LP) print service

The following course map shows how this module fits into the current instructional goal.

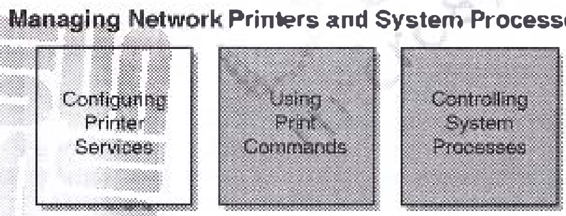

Figure 12-1 Course Map

Introducing Network Printing Fundamentals

The Solaris™ Operating Environment (Solaris OE) LP print service provides a complete printing environment that allows the sharing of printers across systems and a set of software utilities that enable users to print files while they continue to work on other tasks.

Print Management Tools

The LP print service software contains the following components for the set up and administration of printers in the Solaris OE:

- Solaris OE Print Manager – A graphical user interface (GUI) that provides the ability to configure and manage printers.
- LP print service commands – A command-line interface that configures and manages printers. These commands also provide functionality not available in the other print management tools.

Client-Server Model

The Solaris OE print service is implemented in a client-server model.

Print Server

A print server is any system that is configured to manage a printer directly connected to it or that is attached to the network. The print server makes the printers available to other systems on the network and provides spooling for the client's print requests.

Print Client

A print client is a system that sends print requests to a print server.

Types of Printer Configurations

As a system administrator, you must configure printers so that users have access to one or more printers.

You should distribute printers over several print servers. If one print server becomes unavailable, print requests can be quickly and easily routed to other print servers on the network.

The Solaris OE supports local, network, and remote printer configurations.

Local Printer

A local printer is physically connected to a system and is accessed from that system.

Network Printer

A network printer is physically attached to the network and has its own host name and Internet Protocol (IP) address. A network printer provides print services to clients but is not directly connected to a print server.

Remote Printer

A remote printer is one that users access over the network, that is, a printer that is either physically connected to a remote system or physically attached to the network.

Figure 12-2 shows the concept of local, network, and remote printers.

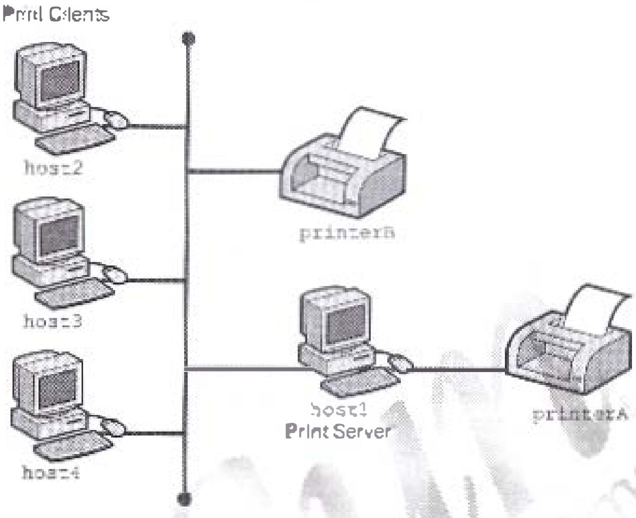

Figure 12-2 Local, Network, and Remote Printers

The printer named printerA, connected to the system named host1, is a local printer for any user logged on to that system.

The printer named printerB is a network printer that is controlled by the print server, host1. This is a network printer for any users logged in on host1, host2, host3, or host4.

For users who are logged in to host2, host3, or host4, both printerA and printerB can be accessed as remote printers.

Basic Functions of the Solaris OE LP Print Service

Basic functions of the Solaris OE LP print service include initialization, queuing, tracking, fault notification, and filtering.

Initialization

The Solaris OE LP print service initializes a printer prior to sending it a print request. The initialization function ensures that the printer is in a known state.

Queuing

The Solaris OE LP print service queues the print requests. The queuing function schedules the print requests that are waiting to be sent to the printer.

Tracking

The Solaris OE LP print service tracks the status of every print request. The tracking function enables the root user to manage all of the requests and typical users to view or cancel their own requests. This function also logs any errors that have occurred during the printing process.

Fault Notification

The Solaris OE LP print service provides fault notification if a problem occurs in the print service. The fault notification function prints an error message on the console or sends an email to the root user, depending on how the service has been configured.

Filtering

The Solaris OE LP print service provides filtering capabilities that convert print jobs to the appropriate type of file for the destination printer.

LP Print Service Directory Structure

The Solaris OE LP print service includes a directory structure, files, and logs. The following section describes some of the more important components of this structure.

See Figure 12-3 for an example of the LP print service directory.

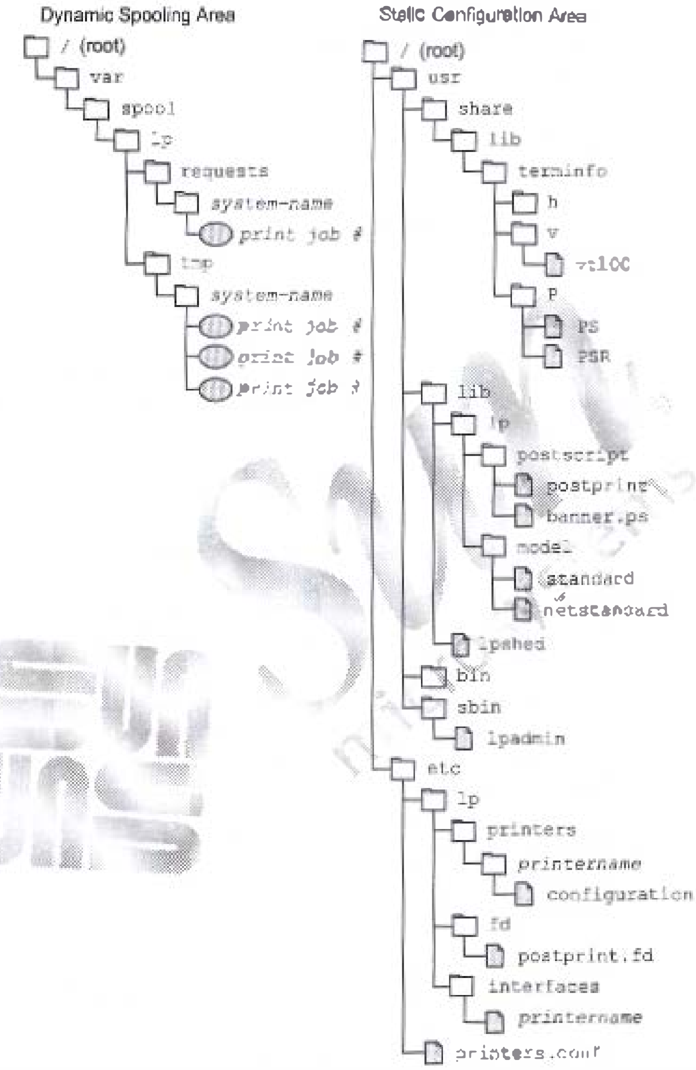

Figure 12-3 LP Print Service Directory Structure

The /usr/bin Directory

This directory contains the LP print service user commands, such as the lp, lpstat, and cancel commands.

The /usr/sbin Directory

This directory contains the LP print service administrative commands, such as the lpadmin, lpusers, and lpshut commands.

The /usr/share/lib/terminfo Directory

This directory contains the terminfo database directories, which describe the capabilities of printers and terminals.

The /usr/lib/lp Directory

This directory contains the lpsched daemon, binary files that the LP print service uses, PostScript™ filters, and standard printer interface programs. Two important subdirectories in the /usr/lib/lp directory are the model and postscript directories.

The /usr/lib/lp/model Directory

This directory contains two default printer interface programs or shell scripts, called the standard and the netstandard scripts.

The standard script supports local printers. For example, when a print request is queued for printing, the print service runs the printer's standard script to:

- Initialize the printer port, if necessary
- Initialize the actual printer, using the terminfo database to find the appropriate control sequences
- Print a banner page, if necessary
- Print the correct number of copies, as specified by the user's print request

The `netstandard` script specifically supports network printers. It collects the spooler and print database information needed to perform network printing and passes the information to a print output module. The `netpr` module opens the network connection to the printer and sends the data to the printer.

Note – The `netpr` module is located in the `/usr/lib/lp/bin` directory.

The `root` user can modify any printer's interface script. For example, to turn off the printing of a banner page, edit the `/etc/lp/interfaces/printer_name` file on the print server. Change the `nobanner` line from:

```
nobanner="no"
```

to

```
nobanner="yes"
```

The `/usr/lib/lp/postscript` Directory

This directory contains all PostScript filter programs provided by the Solaris OE LP print service.

Note – Print filters are programs that the print server uses to convert the content type of a queued print request from one format to another format that is acceptable to the destination printer.

The PostScript print filters in this directory handle many situations in which the printer requires the content of files to be in PostScript format.

These filters have companion descriptor files in the `/etc/lp/fd` directory that tell the LP print service the characteristics and location of the filters.

The `/etc/lp` Directory

This directory contains a hierarchy of LP server configuration directories and files.

You can view the contents of these configuration files. However, you should not edit these files directly. To make configuration changes, use the `lpadmin` command or `printmgr` GUI.

There are three subdirectories in the /etc/lp directory that are important to printer configuration. These are the fd, interfaces, and printers directories.

- The /etc/lp/fd directory contains a set of print filter descriptor files. These files describe the characteristics of the filter and point to the actual filter program.

Note – The /etc/lp/filter.table file contains a filter lookup table.

- The /etc/lp/interfaces directory contains each printer's interface script file. When a printer is configured, the print service places a copy of the appropriate default interface script from the /usr/lib/lp/model directory into the /etc/lp/interfaces/*printername* file. The *printername* variable is the file created that contains the newly configured printer's own interface script.
- The /etc/lp/printers directory contains a subdirectory for each printer served by the system. Each subdirectory contains configuration information and alert files for an individual printer.

  For example, the configuration file for a printer named printerB can contain the following information:

```
# cat /etc/lp/printers/printerB/configuration
Banner: optional
Content types: postscript
Device: /dev/null
Interface: /usr/lib/lp/model/netstandard
Printer type: PS
Modules:
Options: dest=printerB,protocol=bsd
```

The /var/spool/lp Directory

This directory contains a list of current requests that are in the print queue.

The lpsched daemon for each system keeps track of print requests in the following directories:

- /var/spool/lp/tmp/*system-name*
- /var/spool/lp/requests/*system-name*

With a local print request, the `/var/spool/lp/tmp/system-name` directory contains one file, and the `/var/spool/lp/requests/system-name` directory contains another file.

With a remote print request, the `/var/spool/lp/tmp/system-name` directory contains two files, and the `/var/spool/lp/requests/system-name` directory contains one file.

Only the `root` user or `lp` users can access the information in the `/var/spool/lp/requests/system-name` directory.

Only the user who submitted the print request, the `root` user, or the `lp` user can access the information in the `/var/spool/lp/tmp/system-name` directory.

These files remain in their directories only as long as the print request is in the queue. After completing the print request, the print service combines the information in the files and appends it to the `/var/lp/logs/requests` file.

Note – The `/var/spool/print` directory contains the client-side request staging area for the LP print service.

The `/var/lp/logs` Directory

This directory contains an ongoing history of print requests. The log file `/var/lp/logs/requests` contains information about completed print requests that are no longer in the print queue.

The `/usr/sbin/inetd` Internet Service Daemon

The Internet services daemon, `inetd`, is the server process for many network services. It is usually started up at system boot time. The daemon listens for service requests on the ports that are associated with each of the services listed in its configuration file, `/etc/inetd.conf`. When a request arrives, the `inetd` daemon executes the server program that is associated with the service. Print servers listen for print requests with the `inetd` daemon, and upon hearing a request, start up the `in.lpd` daemon.

The /usr/lib/print/in.lpd Program

The inetd daemon starts the in.lpd program, sometimes referred to as the protocol adapter. The in.lpd program implements the network listening service for the print protocol. The print protocol provides a remote interface that enables systems to interact with a local spooling system. This protocol defines standard requests from the print client to the print server, such as requests to start queue processing, to transfer print jobs, to retrieve print status, and to cancel print jobs.

Upon the receipt of a connect request, the in.lpd program starts and services the connection. The in.lpd program closes the connection and exits after servicing the request.

The /usr/lib/lp/lpsched Daemon

The LP print service has a scheduler daemon called lpsched. The scheduler daemon updates the LP system files with information about printer setup and configuration. It also manages requests issued to the system by the lp and lpr commands.

The lpsched daemon schedules all of the local print requests on a print server. It also tracks the status of printers and filters on the print server. When a printer finishes a request, the lpsched daemon schedules the next request, if there is one in the queue on the print server.

Each print server has by default only one lpsched daemon running. It is started by the control script /etc/rc2.d/S80lp when the system is booted (or enters run level 2). The parent lpsched daemon spawns a child lpsched processes to service print jobs.

Solaris OE Printing Process

Users submit print requests from print clients by using the `lp` or `lpr` commands.

Note – The Solaris OE Print Service accepts both the System V Interface Definition (SVID) `/usr/bin/lp` command and the Berkeley Software Distribution (BSD) `/usr/ucb/lpr` command to submit print requests.

Users should use these commands to print text files. These commands do not print documents created in applications such as FrameMaker. Most third-party applications require you to print from a selection menu within the application.

The function of the `lp` and `lpr` commands is to queue print requests for printing on a destination printer.

Locating the Destination Printer

The Solaris OE LP print service checks several resources to locate the destination printer for a print request.

Figure 12-4 shows the resources checked as it identifies the appropriate printer for a print request.

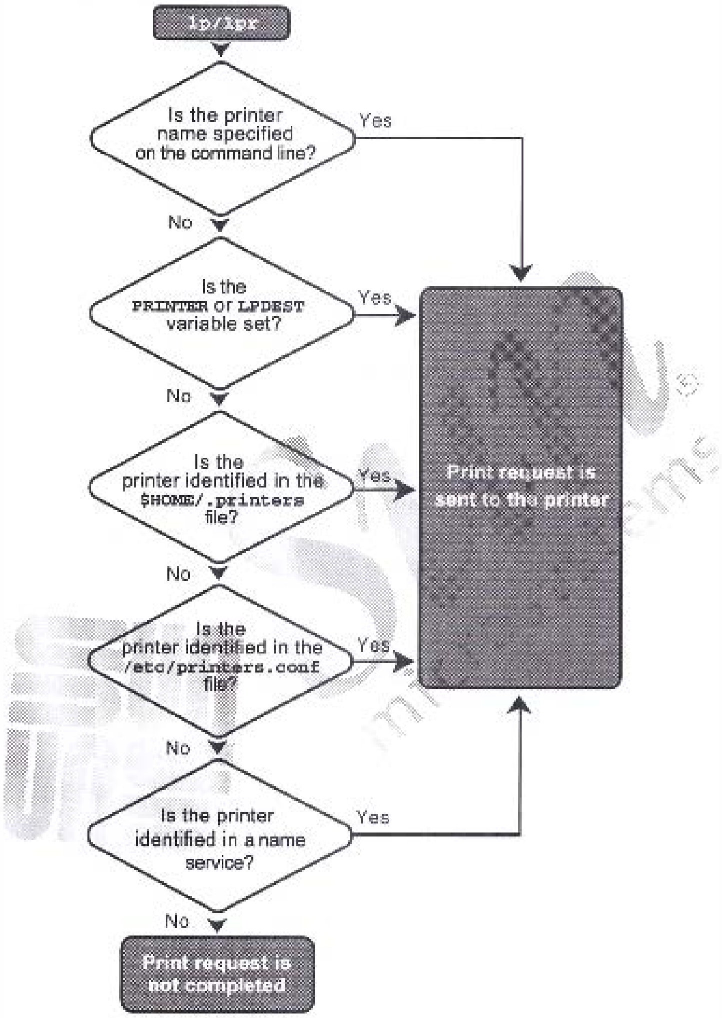

Figure 12-4 Locating the Destination Printer

If the command line does not specify a named printer destination, the user's shell environment is checked.

You can set the LPDEST or PRINTER environment variables to a default printer name. The lp command checks LPDEST and then PRINTER. The lpr command reverses the order when searching for a printer.

If neither variable specifies a named printer destination, then the Solaris OE LP print service checks for the variable named _default in the following files:

- The $HOME/.printers file

  Users can create their own .printers file in their home directory to set the default printer name. They should add the following line to the file:

  _default printername

  If the $HOME/.printers file does not exist or does not specify a printer name destination, then the Solaris OE LP print service checks the /etc/printers.conf file.

- The /etc/printers.conf file

  Each entry in the /etc/printers.conf file describes a printer destination. For example, if host1 is the print server's name and printerA is the printer's name, the entry in this file appears as follows:

```
_default:\
        :use=printerA:\
printerA:\
        :bsdaddr=host1,printerA,Solaris
        :description=printerA
```

If the _default variable is not set, then the _default variable in the name service database (for example, Network Information Service (NIS)) is checked.

- The printers.conf.byname file

  The printers.conf.byname file is the NIS version of the /etc/printers.conf file. In this case, the _default variable entry in the name service map called printers.conf.byname defines the print server and printer name destination:

  _default:bsdaddr=servername,printername:

If the destination printer name cannot be located in any of these configuration resources, the print request cannot be completed.

Note – The last three files described in the following paragraphs rely on the `printers:` entry in the NIS version of the `/etc/nsswitch.conf` file.

An example of the `/etc/nsswitch.conf` file syntax is:

```
printers: user files nis
```

where:

`user` = Checks `$HOME/.printers` file
`files` = Checks `/etc/printers.conf` file
`nis` = Checks `printers.conf.byname` file

Introducing the Local Print Process

When a user submits a print request to a local printer, the `lp` or `lpr` command sends the request to the `lpsched` daemon. The `lpsched` daemon is also called the print scheduler.

Figure 12-5 shows the role of the lpsched daemon in the printing process.

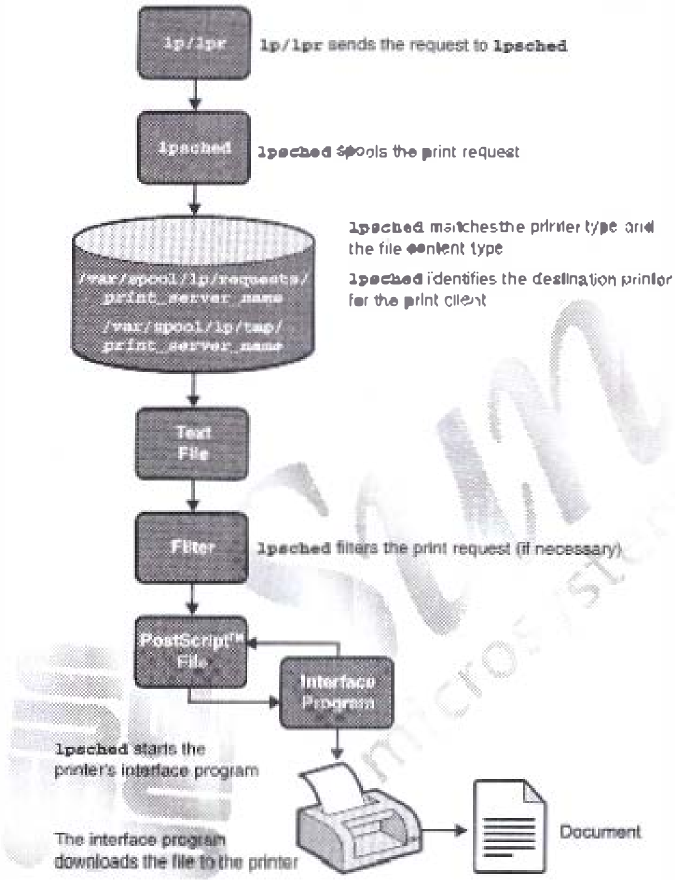

Figure 12-5 Local Printing Process

The lpsched daemon matches the printer type and identifies the default printer for the system. It then filters the print job.

The lpsched daemon keeps track of print requests in the following directories:

- /var/spool/lp/requests/*system_name*
- /var/spool/lp/tmp/*system_name*

If the printer is free, the lpsched daemon starts the printer's interface program. The interface program performs the following functions:

- Initializes the printer port
- Initializes the printer
- Prints the banner page
- Prints the correct number of file copies
- Sends any fault notifications

Remote Print Process

When a user submits a print request to a remote printer, the lp or lpr command sends the print request directly to the print server.

The print server processes the print request and sends the print request to the destination printer to be printed.

Figure 12-6 shows a remote print request submitted from a print client to a print server in the Solaris OE.

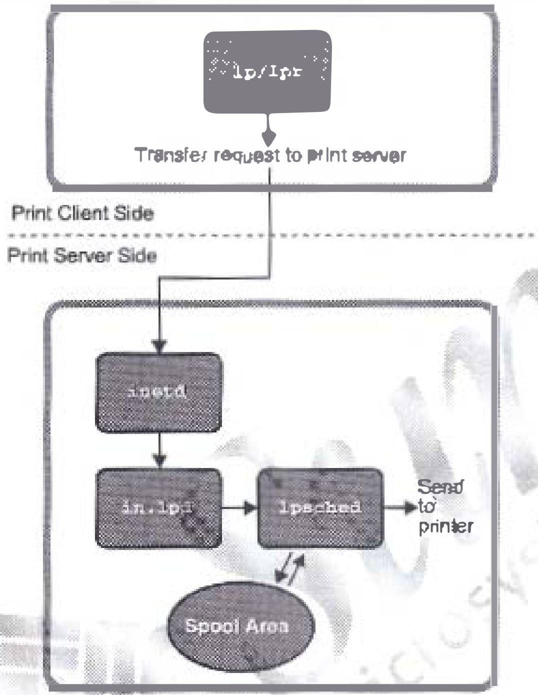

Figure 12-6 Solaris OE Remote Printing

The client's print command communicates directly with the print service on the server to transfer a print request to the printer.

The print server listens for print requests with the Internet services daemon inetd. When the inetd daemon hears a request for a print service on the network, it starts the in.lpd program. The in.lpd program is also called the print protocol adapter. The in.lpd program starts on demand and exits when the network request finishes.

The print protocol adapter translates the print request, communicates it to the print spooler, and returns the results to the print requester.

The print protocol adapter contacts the lpsched daemon to start the printer's interface program and to transfer the print request to the destination printer.

Configuring Printer Services

Configuring printer services in the Solaris OE involves a number of key tasks. Table 12-1 shows these tasks.

Table 12-1 Main Tasks for Configuring Printer Services

Tasks	Description
Setting up the printer	Physically connecting the printer to a system or the network
Setting up the print server	Configuring the system that is to manage and provide access to the printer
Setting up the print client	Configuring the system to access a remote printer
Verifying printer access	Checking that the print server recognizes all print clients and that each print client recognizes the print server

Note – When a network of systems is not running a name service, such as NIS, enter each print server's host name and IP address in the /etc/inet/hosts file on the print client when you are setting up the printer services.

Identifying Print Server Requirements

Any system on the network can be a print server if it has the resources to manage the printing load, such as spooling space and memory.

Spooling Space

The spooling space is the amount of disk space that is used to store and process print requests. Spooling space is the most important factor to consider when designating systems as print servers. The recommended starting size for spooling space is from 25 to 500 Mbytes, depending on the type and the size of files being printed and the number of users.

Note – The term spool is an acronym for system peripheral operation offline.

Memory

The Solaris OE requires 64 Mbytes of memory to run on a system. Print servers do not require additional memory. However, an extra 32 Mbytes of memory can improve performance when the server is filtering print requests.

Using the Solaris OE Print Manager

The Solaris OE Print Manager enables you to set up and manage printers.

The Solaris OE Print Manager is the preferred method for managing printers. When used with a name service such as NIS, it centralizes printer information and simplifies printer administration.

Note – The Solaris OE Print Manager recognizes existing printer information on print servers, print clients, and in the name service databases.

The following steps demonstrate how to configure a network printer with the Solaris Print Manager. As the root user, start the Solaris OE Print Manager with the following command:

```
# /usr/sadm/admin/bin/printmgr &
```

You can also start the Solaris OE Print Manager by selecting the Printer Administrator from the Tools option on the Common Desktop Environment (CDE) Workspace menu and entering the host name of the workstation to continue.

Either method displays the Solaris OE Print Manager main window, with Figure 12-7 overlaid on top of it.

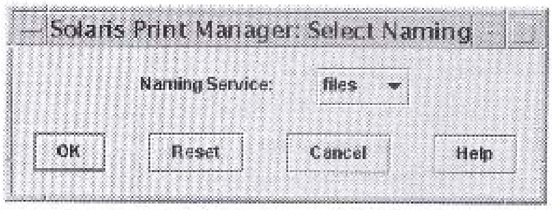

Figure 12-7 Select Naming Service Window

1. Click OK to select the default, `files`.

   Figure 12-8 remains on the screen.

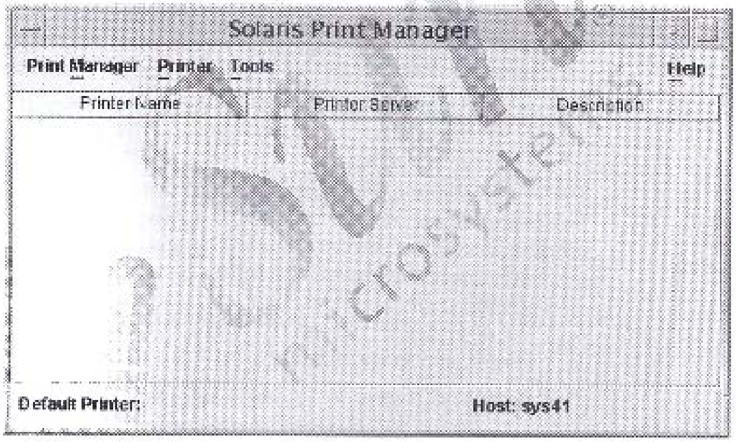

Figure 12-8 Solaris OE Print Manager Window

2. Click the Printer menu in this window. Figure 12-9 shows possible menu selections on the Printer menu.

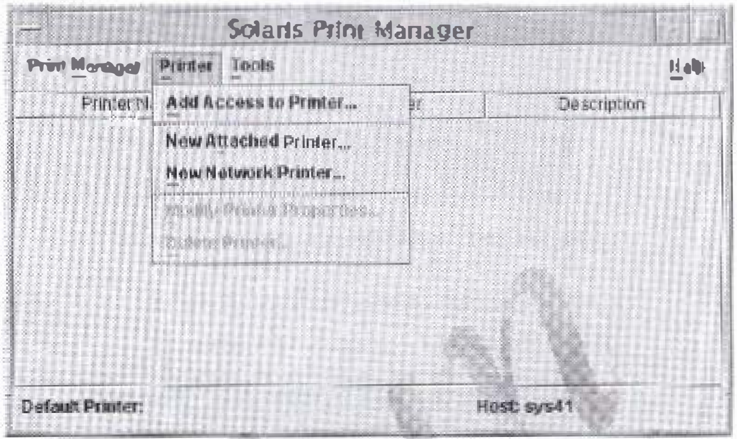

Figure 12-9 Solaris OE Print Manager Printer Menu

Note – By clicking Print Manager and selecting Show Command Line Console, you can see the command-line equivalents to each of the actions taken to configure printers. You can then save these steps as commands to perform similar actions in the future or build your own scripts for configuring printers.

Menu selections include:

- Add Access to Printer – Selected from a print client to set up access to printers that are controlled by a print server. The host name and IP address of the print server must be in the print client's `/etc/inet/hosts` file or in a name service database (for example, NIS).
- New Attached Printer – Selected from a print server to configure a printer that is physically connected to it. The print server provides the queuing capabilities, filtering, and printing administration.
- New Network Printer – Selected from a print server to configure a printer that is directly attached to the network. The print server provides the queuing capabilities, filtering, and printing administration. The network printer's name and its IP address must be entered either in the print server's `/etc/inet/hosts` file or in a name service database.

Configuring a New Network Printer

Table 12-2 shows the information you would use to configure the new network printer.

Table 12-2 Information Fields for Configuring a New Network Printer

Required Field	Description
Printer Name	A unique name for the network printer. The name can contain a maximum of 14 alphanumeric characters, including dashes and underscores. This is the name entered on the command line with a print command.
Printer Server	Defaults to the name of the system on which you are currently running the Solaris OE Print Manager. This system is the print server for this network printer.
Description	This field is optional. A printer's description commonly contains information to help users identify the printer (for example, physical location or printer type).
Printer Type	The generic name for the type of printer (for example, PostScript, HP Printer, Diablo). The LP print service identifies each printer by its printer type. Printer type data is held in the directory /usr/share/lib/terminfo. The Other option, located at the end of the list, allows for the selection of any other printer type listed in the terminfo database.
File Contents	Specifies the data format of files that can be printed without any special filtering by the LP print service software.
Fault Notification	The list of choices for how the superuser is notified of printer errors. These include: Write to Superuser, Mail to Superuser, or None.

Table 12-2 Information Fields for Configuring a New Network Printer (Continued)

Required Field	Description
Destination	The network printer's unique access name. The Destination access name can be either the name of the printer or its IP address as defined in the /etc/inet/hosts file or in a name service database. The Destination access name is used only by the print subsystem when it is making the network connection to the physical printer or the printer-host device. It becomes part of the printer configuration database and is associated with the network printer's IP address.
Protocol	The internet protocol that is used to communicate with the printer for file transfer. The choices are Berkeley BSD Printer Protocol and raw Transmission Control Protocol (TCP). In general, the TCP protocol is more generic across printers. The printer vendor documentation supplies the information about the protocol to select.
Options	Identifies two options, the Default Printer option and the Always Print Banner option, which, by default, are disabled. To enable an option, click in the appropriate box (a check mark appears).
User Access List	Specifies print clients that can print to this printer. By default, the word all allows every print client access to this printer.

From the print server, use the following procedure to set up the configuration information to provide access to a new network printer.

3. From the Printer menu, select the New Network Printer option.

   Figure 12-10 shows the window that appears.

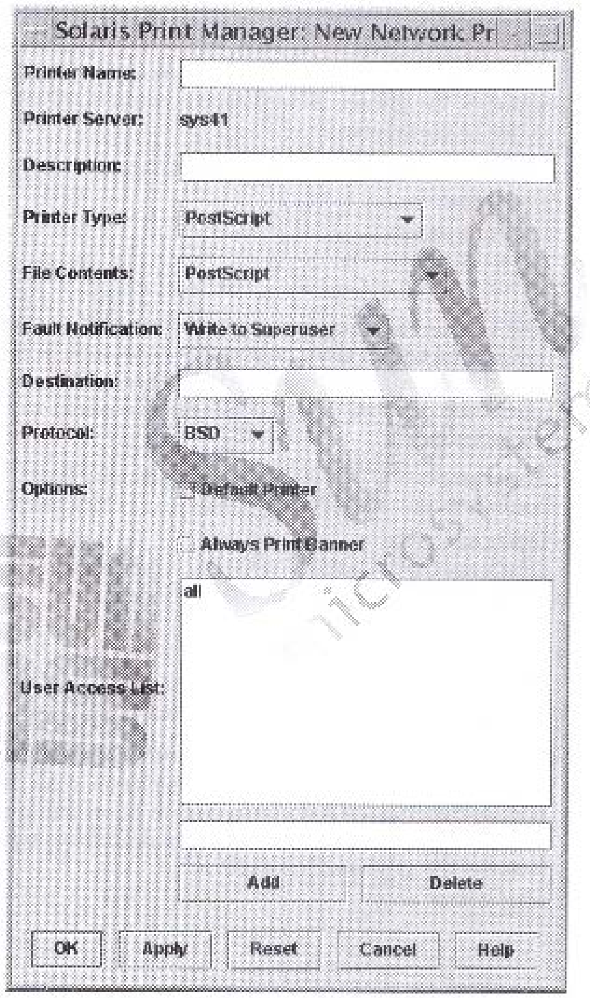

Figure 12-10 Solaris Print Manager: New Network Printer Window

4. In the Printer Name field, enter the new printer name, for example, printerA.
5. Click the Description field, and enter a printer description of your choice.
6. For the purposes of this demonstration, accept the default Printer Type: PostScript.

   The LP print service uses information in the terminfo database to initialize the printer, as well as to communicate the sequence of codes to the printer.

   To view the contents of the terminfo directory, type the following command:

```
# ls /usr/share/lib/terminfo
1  2  3  4  5  6  7  8  9  A  a  B  b  c  d  e  f  G  g  H  h  i  j  k  l
M  m  n  o  P  p  q  r  S  s  t  u  v  w  x  y  z
```

   The terminfo directory contains many different subdirectories that are named with a letter or digit. Use the same initial letter or digit that the manufacturer assigned to the printer's generic name. The terminfo database includes information about terminals and modems too.

   For example, the printer type for a particular Epson printer would be located in the subdirectory /usr/share/lib/terminfo/e.

```
# ls /usr/share/lib/terminfo/e
emots            ep2500+high      ep48             ergo4000         exidy2500
env230           ep2500+low       epson2500        esprit
envision230      ep40             epson2500-80     ethernet
ep2500+basic     ep4000           epson2500-hi     ex3000
ep2500+color     ep4080           epson2500-hi80   exidy
```

7. Accept the default File Contents: PostScript.

   Every printer has configuration information pertaining to the content type of files that it can accept for its printer type. The LP print service depends on this configuration information to match the content type of each print request to the printer type, which ensures that the file is printed correctly.

By selecting a file content ty e, as shown in Table 12-3, you can specify the data format of the file that can be printed without any special filtering by the print software.

Table 12-3 Descriptions of File Content Types

File Content Type	Description
ASCII	ASCII files do not require filtering.
PostScript	PostScript files do not require filtering. PostScript is the default.
Both PostScript and ASCII	PostScript and ASCII files do not require filtering.
None	All files require filtering, except those matching the printer's type.
Any	No filtering required. If printer cannot handle the file content type, the file is not to be printed.

8. Click Fault Notification, nd select the Mail to Superuser option.
9. Click the Destination field, and type a Destination access na me.

   If the network printer is not recognized by it name or IP address in the hosts table, you might need to use the vendor-supplied access name fot the network printer, which is *sometimes* qualified by a designated port number. These are both explicitly defin d in the printer vendor's documentation.

Table 12-4 shows the format for a Destination entry.

Table 12-4 Destination Entry Format

Destination	Protocol
printer_name	BSD
system_name:printer	BSD
IP_ADDR	BSD
IP_ADDR:port_number[1]	TCP
printer_node_name:port_number	TCP

1. The port number is print server dependent. For example, LexMark uses Port 9100.

10. Leave the Internet protocol set to BSD.
11. Click in the Default Printer box to enable the Default Printer option.

Note – If enabled, the Default Printer option designates this printer as the default printer for print jobs from this system.

---

12. You can (optionally) click in the Always Print Banner box to enable the Always Print Banner option.
13. Accept the default, all, for the User Access List. This allows all users on all systems to use the printer.

To restrict user access to this printer, you can enter the values shown in Table 12-5 in the text field below the User Access List window.

Table 12-5 User Access Values

Value	Definition
user-name	The specified user, for example `user1`, can access the printer from any system.
system-name`!`*user-name*	The specified user from the named system can access the printer, for example, `host2!user4`.
system-name`!all`	All users from the named system only can access the printer, for example, `host5!all`.
`all!`*user-name*	The specified user from all systems can access the printer, for example, `all!user1`.

Note – To delete an entry from the User Access List, select the entry, and click **Delete**.

14. To accept the new network printer's configuration information, click **OK**.

    Figure 12-11 shows the Solaris OE Print Manager window, which is displaying the newly configured printer.

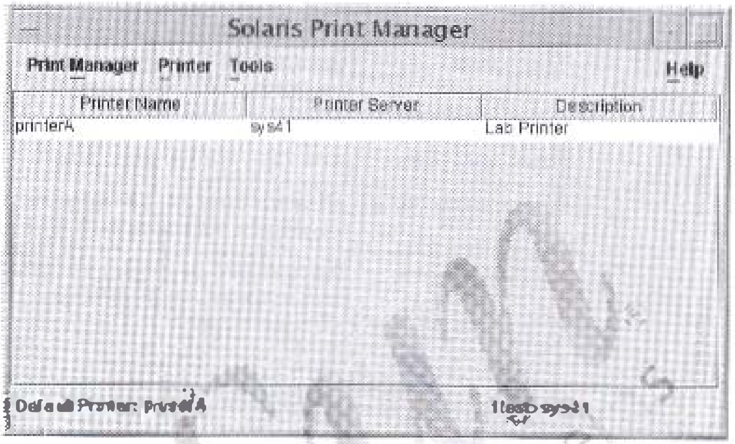

Figure 12-11 Solaris OE Print Manager Window: Configured Printer

15. To close the Solaris OE Print Manager window, select the **Exit** option from the Print Manager menu.

Administering Printer Services

You use the lpadmin command to configure the LP print services from the command line.

You could use this command to perform the following tasks:

- Defining printer devices and printer names
- Specifying interface programs (custom or standard) and printer options
- Defining printer types and file content types
- Creating printer classes
- Defining allow and deny user lists
- Specifying fault recovery
- Removing printers and printer classes

The lpadmin command is most commonly used by the root user for the purpose of:

- Creating printer classes
- Setting or changing a system's default printer destination
- Removing a printer's configuration from the LP print service

Configuring Printer Classes

You can increase printer access by establishing printer classes. A printer class is a way of grouping individual printers so that they can be identified by a single name known as a class name.

After a printer class is created, you use it as the destination for users' print requests. The LP print service automatically sends each print request to the first available printer within the class that matches the content type expected by the printer. This useful feature can help you balance the load of print requests among several printers.

A printer class can include:

- Specific printer types (for example, all PostScript printers)
- Printers in a specific location (for example, Building 2)
- Printers in a specific work group or department (for example, Marketing, Engineering, Accounting).

You can create a printer class by using the lpadmin command only on the print server for which the printers are configured. Printer classes cannot be defined on print clients.

Configuring Printer Priority Within a Class

When you create a printer class, the root user can control the printer access order by adding the printers to the class in a descending order. For example, by adding a high-speed printer to the printer class first, you can enable it to handle as many print requests as possible, before off-loading to the printer that was added to the class next, and so on.

Creating a Printer Class

You create a printer class when the first printer is added to the printer class name. After creating a printer class, you can add other printers to it at any time.

The following example creates a printer class called bldg2:

```
# lpadmin -p printerB -c bldg2
```

The following example adds another printer (printerD) to this class:

```
# lpadmin -p printerD -c bldg2
```

After you have finished adding printers to the printer class, use the accept command to allow queuing of print requests to the new print queue (bldg2 in the example).

Note – The accept command is explained in Module 13, "Using Print Commands."

```
# accept bldg2
destination "bldg2" now accepting requests
```

Use the lpstat -t command on the print server to check the status of the new printer class:

```
# lpstat -t
scheduler is running
system default destination: printerA
members of class bldg2:
      printerB
      printerD
device for printerB: /dev/null
device for PrinterD: /dev/null
bldg2 accepting requests since Fri Jan  4 10:37:44 MST 2002
printerB accepting requests since Fri Jan  4 10:37:44 MST 2002
printerD accepting requests since Fri Jan  4 10:37:44 MST 2002
```

To send a print request to a printer class, perform the following command:

```
# lp -d bldg2 myfile
request id is bldg2-0 (1 file)
```

Setting the System's Default Printer

The root user can run the lpadmin command to set an individual printer or a printer class to be the system's default destination for all print requests.

```
# lpadmin -d printername
# lpadmin -d printer-classname
```

For example, to set a system's default destination printer, perform the command:

```
# lpadmin -d printerE
```

To verify that the system's default destination printer has been set, perform the command:

```
# lpstat -d
system default destination: printerE
```

To verify an individual user's default destination printer, perform the command:

```
$ lpstat -d
system default destination: users_printer
```

The print request issued is sent by default to printerE.

```
# lp myfile
request id is printerE-514 (1 file)
```

Changing the System's Default Printer Class

To change a system's default destination printer to the class bldg2, perform the command:

```
# lpadmin -d bldg2
#
# lpstat -d
system default destination: bldg2
```

Note – You cannot activate or deactivate a printer class with the enable and disable commands. You can activate or deactivate only the individual printers within a printer class. But you can allow or disallow spooling a classes' jobs by using the commands accept and reject. The commands enable, disable, accept, and reject are explained in Module 13, "Using Print Commands."

Removing a Client's Printer Configuration

To remove a printer's configuration manually on the client side, perform the following:

1. Log in as the root user on the print client that has access to the printer to be removed from the LP print service.
2. Delete information about the printer from the print client by performing an lpadmin command.

```
# lpadmin -x printername
```

where -x deletes the specified printer.

For example, the following command deletes printerD from the system.

```
# lpadmin -x printerD
```

Information for the specified printer is deleted from the print client's /etc/printers.conf file.

Repeat Steps 1 and 2 for each print client that has access to the printer.

Removing a Server's Printer Configuration

Note – The reject and disable commands are explained in Module 13, "Using Print Commands."

To remove a printer's configuration manually on the server side, perform the following:

1. Log in as the root user on the print server on which the printer is configured.
2. Stop queuing print requests on the printer.

```
# reject printerD
```

3. Stop the printer.

```
# disable printerD
```

4. Delete the printer from the print server.

```
# lpadmin -x printerD
```

This action deletes configuration information for the printer from the print server's /etc/lp/printers directory and /etc/printers.conf file.

Starting and Stopping the LP Print Service

The LP print service is started by the lpsched daemon and is shut down by the lpshut command.

Starting the LP Print Service

The lpsched daemon starts or restarts the LP print service. Printers that are restarted with a lpsched command from the command line, reprint, in their entirety, the print requests stopped by the lpshut command.

The following is an example of starting the lpsched daemon from the command line:

```
# /usr/lib/lpsched
Print services started
```

The lp print service script, located in the /etc/init.d directory, also can be used to start the lpsched daemon.

```
# /etc/init.d/lp start
Print services started
```

Stopping the LP Print Service

The lpshut command stops the LP print service. Any printers that are currently printing when the command is invoked stop printing.

The following is an example of the lpshut command:

```
# /usr/lib/lpshut
Print services stopped.
```

The lp print service script, located in the /etc/init.d directory, also can be used to stop the lpsched daemon.

```
# /etc/init.d/lp stop
Print services stopped.
```

Module 13

Using Print Commands

Objectives

Upon completion of this module, you should be able to:

- Specify a destination printer
- Use the LP print service

The following course map shows how this module fits into the current instructional goal.

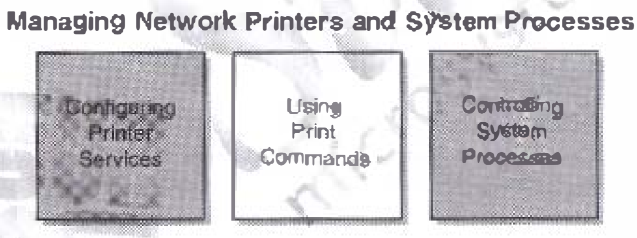

Figure 13-1 Course Map

Specifying a Destination Printer

In the Solaris OE, users submit print requests by using the `lp` command or the `lpr` command.

Note – The Solaris OE LP Print Service accepts both the SVID `/usr/bin/lp` command and the BSD `/usr/ucb/lpr` command to submit print requests.

Using the `lp` Command

The `lp` command is located in the `/usr/bin` directory. The `lp` command submits a print job to the default printer or to another printer (by specifying the printer name). To use the command, perform one of the following commands:

```
$ /usr/bin/lp filename
$ /usr/bin/lp -d printername filename
```

Using the `lpr` Command

The `lpr` command is located in the `/usr/ucb` directory. The `lpr` command functions in the same manner as the `lp` command—it submits a print job to the default printer or to another printer.

```
$ /usr/ucb/lpr filename
$ /usr/ucb/lpr -P printername filename
```

The preceding examples of the print commands demonstrate the atomic style. You can also use the Portable Open Systems Interface (POSIX) style to specify a destination printer.

To submit a print request that uses the POSIX style, include the print command and an option, followed by the printer server name, a colon, and the printer name as configured on the print server.

The full command-line entry is as follows:

```
$ /usr/bin/lp -d hostname:printername filename
$ /usr/ucb/lpr -P hostname:printername filename
```

Using the LP Print Service

The LP print service is a set of software commands, utilities, and filters that allow users to print files and the root user to set up and manage the print operations.

Table 13-1 lists some of the more commonly used print service administration commands.

Note – You must be the root user to use these commands.

Table 13-1 LP Print Service Administration Commands

Command Name	Description
accept	Permits print requests to be queued for the specific printers
reject	Prevents print requests from being queued for the specific printers
enable	Activates the specified printers
disable	Deactivates the specified printer
lpmove	Moves print requests from one printer destination to another

Accepting Print Jobs

As the root user, you use the accept command on the print server to permit print requests to be queued on the specified printers.

Using the accept Command

You use the accept command to allow queuing of print requests for the named destinations. A destination specifies the name of a printer or printer class.

The format for the command is:

```
# /usr/sbin/accept destination(s)
```

In the following example, the root user has enabled the queuing of print requests on printerD.

```
# accept printerD
destination "printerD" now accepting requests
```

Rejecting Print Jobs

As the root user, you use the reject command on the print server to prevent print requests from queuing on the specified printers.

Using the reject Command

You use the reject command to prevent print requests from queuing and stop users from submitting requests to the printer queues.

The format for the command is:

```
# /usr/sbin/reject -r "reason" destination(s)
```

The following example shows how to use the option -r "reason" to enter an explanation for the rejection of print requests for a printer. A user can see that text by issuing the lpstat -a or lpstat -t command.

```
# reject -r "Replacing Toner Cartridge" printerD
destination "printerD" will no longer accept requests
```

Enabling Printers

On the print server, as the root user, you can use the enable command to activate the specified printers.

Using the enable Command

The enable command activates the printers, which enables the printing of requests submitted to the print queues.

The format for the command is:

```
# /usr/bin/enable destination(s)
```

The following example shows how to enable printerD.

```
# enable printerD
printer "printerD" now enabled
```

Disabling Printers

On the print server, as the root user, you can use the disable command to deactivate the specified printers.

Using the disable Command

The disable command deactivates printers, which disable them from printing print requests waiting in the print queues.

By default, any requests currently printing on the printer when the disable command is issued are reprinted in their entirety when the printer is enabled again.

The format for the command is:

```
# /usr/bin/disable -c | -W -r "reason" destination
```

Table 13-2 shows the options for the disable command.

Table 13-2 Options for the disable Command

Option	Definition
-c	Cancels the current job and disables the printer. The current job is not printed later.
-W	Waits until the current job is finished before disabling the printer.
-r	Assigns a reason for the disabling of the printer.

The following example shows how to use the disable command with options.

```
# disable -W -r "Printer down for maintenance" printerD
printer "printerD" now disabled
```

Moving Print Jobs

You use the `lpmove` command to move one or all print requests from one printer destination to another printer destination.

Using the `lpmove` Command

The format for the `lpmove` command is:

```
# /usr/sbin/lpmove source_destination target_destination
```

To move one or all print requests by using the `lpmove` command, complete the following steps.

1. Become the `root` user on the print server.
2. Use the `reject` command to prevent any further print requests from being sent to the print queue. This step notifies users that the printer is not accepting requests.

```
# reject -r "PrinterC is down for repairs" printerC
destination "printerC" will no longer accept requests
```

3. Use the `lpstat` command to display the print queue to see how many print requests are to be moved. This step is needed to identify print request identification numbers (IDs) only if selected print requests are going to be moved to another printer.

```
# lpstat -o
printerC-29    sys41!user2    61426    Jan 07  12:30
printerC-30    sys41!user1     9560    Jan 07  12:30
printerC-31    sys42!user2      845    Jan 07  12:30
printerC-32    sys42!user2      845    Jan 07  12:30
printerC-33    sys42!user2      845    Jan 07  12:30
```

4. Use the `lpstat` command to verify that the destination printer is accepting print requests.

```
# lpstat -a printerA
printer printerA accepting requests since Tue Jan 1
```

5. Move the print requests.

   a. For example, to move all print requests from `printerC` over to `printerA`, perform the following command:

```
# lpmove printerC printerA
move in progress ...
total of 5 requests moved from printerC to printerA
```

b. For example, to move one or more individual print requests from printerC to printerA, perform the following command:

```
# lpmove printerC-32 printerC-33 printerA
total of 2 requests moved to printerA
```

6. When printerC is available again, use the accept command to print jobs to queue to printerC.

```
# accept printerC
destination "printerC" now accepting requests
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using the LP Print Service (Level 1)

In this exercise, you use the Solaris OE Print Manager to set up a print spooler that sends output to a local terminal window, add access to a remote printer, and use print management commands.

Preparation

The host name and IP address of the system that controls the printer you want to access must exist in the /etc/hosts file. Refer to the lecture notes as necessary to perform the tasks listed.

Tasks

Complete the following tasks:

- Open two terminal windows. Record the pseudo terminal device used by one of them. In the other window, run the Solaris OE Print Manager, and define a local Diablo printer that uses the first window's terminal as its output device. Test the new printer.

  (Steps 1–7 in the Level 2 lab)
- Use the Solaris OE Print Manager to gain access to a printer defined on another system. Test the remote printer.

  (Steps 9–13 in the Level 2 lab)
- Manipulate your Diablo printer to:
  - Disable printer output
  - Queue four files for printing
- List all print jobs
- Cancel two jobs by listing their request IDs
- Cancel the remaining jobs by using their associated user names
- Enable printing again
- Reject print requests and supply a reason
- View the reason
- Accept print requests on the default printer

  (Steps 14–24 in the Level 2 lab)

Exercise: Using the LP Print Service (Level 2)

In this exercise, you use the Solaris OE Print Manager to set up a print spooler that sends output to a local terminal window, add access to a remote printe , and use print management commands.

Preparation

The host name and IP address of the system that controls the printer you want to access must exist in the /etc/hosts file. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Open two terminal windows. Record the pseudo terminal device used by one of them. In the other window, run the Solaris OE Print Manager, and define a local Diablo printer that uses the first window's terminal as its output device. Test the new printer.
- Use the Solaris OE Print Manager to gain access to a printer defined on another system. Test the remote printer.
- Use the following commands to manipulate your Diablo printer:
  - `enable`
  - `disable`
  - `lp`
  - `lpstat`
  - `accept`
  - `reject`
  - `cancel`
- Manipulate your Diablo printer to:
  - Disable printer output
  - Queue four files for printing
- List all print jobs
- Cancel two jobs by listing their request IDs
- Cancel the remaining jobs by using their associated user names

- Enable printing again
- Reject print requests and supply a reason
- View the reason
- Accept print requests on the default printer

Tasks

Complete the following steps:

1. Log in as the root user, and open two terminal windows. In one of the windows, use the tty command to identify the pseudo terminal device that it uses. Use this device name as the port for the new printer. For example, the device name in the following output is /dev/pts/5:

```
# tty
/dev/pts/5
```

   Device name:
2. In the other terminal window, run the Solaris OE Print Manager.
3. In the Select Naming Service panel, verify that files is selected, and click OK. From the Print Manager menu, select the Show Command Line Console option. Position the Command Line Console in a convenient location.
4. From the Printer menu, select the New Attached Printer option.
5. Fill in the fields according to Table 13-3. To name your printer, use a name different from that of your system.

Table 13-3 Configuration Fields

Field	Selection or Entry
Printer name	Your choice.
Description	Your choice.
Printer port	Select the Other option. Enter the device name of the terminal window found in Step 1.
Printer type	Diablo.
File contents	ASCII.
Fault notification	Write to superuser.

Table 13-3 Configuration Fields (Continued)

Field	Selection or Entry
Default Printer	(Select the box.)
Always Print Banner	(Do not select the box.)
User Access List	(No change.)

6. Click OK when you are finished. Notice the command-line entries that appear in the console window.
7. Test your printer configuration by printing the /etc/hosts file to the default printer. Observe the output on the other terminal window.

   You should see the contents of the /etc/hosts file scroll through the other window.
8. From the Printer menu, select the Add Access to Printer option.
9. Fill in the fields according to Table 13-4.

Table 13-4 Configuration Fields

Field	Selection or Entry
Printer name	Enter the name of a printer on another system.
Print server	Enter the name of the system on which the preceding printer is defined. Ensure this system name and IP address are in your /etc/hosts file.
Description	Your choice.
Default printer	Do not select the box.

10. Click OK when you are finished.

    Notice the command-line entries that appear in the console window.
11. Test your new configuration by printing the /etc/hosts file to the remote printer. Observe the output on the other system.

    You should see the contents of the /etc/hosts file scroll through the other window.
12. In an available terminal window, use the lpstat command to display the current status information of the printers on your system.
13. Disable print output for your default printer.

14. Send the following four files to your default printer: /etc/hosts, /etc/inittab, /etc/dfs/dfstab, and /etc/skel/local.profile.
15. Check the print queue to find the request ID for each job.

    The four print jobs should be listed with sequential numbers.
16. Use the request IDs to cancel two of the requests. Verify the result. Use the following syntax to cancel the requests:

```
# cancel printername-# printername-#
```

    Two of the print jobs should be gone.
17. Cancel the other two jobs by indicating the user who sent them. Verify the result.
18. Enable printing for your default printer. Use the following syntax:

```
# enable printername
```

19. Set your default printer to reject requests, and display a reason for doing so. For example:

```
# reject -r "Printer is down for maintenance" printername
```

20. Attempt to send a job to the default printer. Observe the messages displayed.

```
# lp /etc/hosts
```

    Your message should say printername: Requests are not being accepted
21. Use the lpstat command to display the reason that the printer is not accepting requests. Use the following syntax:

```
# lpstat -a printername
```

    Your message should say *printername: your reason from step 20.*
22. Set your default printer to again accept requests.

```
# accept printername
```

Exercise: Using the LP Print Service (Level 3)

In this exercise, you use the Solaris OE Print Manager to set up a print spooler that sends output to a local terminal window, add access to a remote printer, and use print management commands.

Preparation

The host name and IP address of the system that controls the printer you want to access must exist in the /etc/hosts file. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Open two terminal windows. Record the pseudo terminal device used by one of them. In the other window, run the Solaris OE Print Manager, and define a local Diablo printer that uses the first window's terminal as its output device. Test the new printer.
- Use the Solaris OE Print Manager to gain access to a printer defined on another system. Test the remote printer.
- Use the following commands to manipulate your Diablo printer:
  - `enable`
  - `disable`
  - `lp`
  - `lpstat`
  - `accept`
  - `reject`
  - `cancel`
- Manipulate your Diablo printer to:
  - Disable printer output
  - Queue four files for printing
- List all print jobs
- Cancel two jobs by listing their request IDs
- Cancel the remaining jobs by using their associated user names

- Enable printing again
- Reject print requests and supply a reason
- View the reason
- Accept print requests

Tasks and Solutions

Complete the following steps:

1. Log in as the root user and open two terminal windows. In one of the windows, use the tty command to identify the pseudo terminal device it uses. Use this device name as the port for the new printer. For example, the device name in the following output is /dev/pts/5:

```
# tty
/dev/pts/5
```

   Device name: *Your device name will vary.*

2. In the other terminal window, run the Solaris OE Print Manager:

```
# /usr/sadm/admin/bin/printmgr &
```

3. In the Select Naming Service panel, verify that files is selected, and click OK. From the Print Manager menu, select the Show Command Line Console option. Position the Command Line Console in a convenient location.
4. From the Printer menu, select the New Attached Printer option.
5. Fill in the fields according to Table 13-3 on page 13-11. To name your printer, use a name different from that of your system.
6. Click OK when you are finished. Notice the command line entries that appear on the console window.
7. Test your printer configuration by printing the /etc/hosts file to the default printer. Observe the output on the other terminal window.

```
# lp /etc/hosts
```

   You should see the contents of the /etc/hosts file scroll through the other window.

8. From the Printer menu, select the Add Access to Printer option.
9. Fill in the fields according to Table 13-4 on page 13-12.

10. Click OK when you are finished.

    Notice the command line entries that appear in the console window.

11. Test your new configuration by printing the /etc/hosts file to the remote printer. Observe the output on the other system.

```
# lp -d printername /etc/hosts
```

You should see the contents of the /etc/hosts file scroll through the other window.

12. In an available terminal window, use the lpstat command to display the current status information of the printers on your system.

```
# lpstat -t
```

13. Disable print output for your default printer.

```
# disable printername
```

14. Send the following four files to your default printer: /etc/hosts, /etc/inittab, /etc/dfs/dfstab, and /etc/skel/local.profile.

```
# lp /etc/hosts
# lp /etc/inittab
# lp /etc/dfs/dfstab
# lp /etc/skel/local.profile
```

15. Check the print queue to find the request ID for each job.

```
# lpstat -o
```

The four print jobs should be listed with sequential numbers.

16. Use the request IDs to cancel two of the requests. Verify the result. Use the following syntax to cancel the requests:

```
# cancel printername-# printername-#
# lpstat -o
```

Two of the print jobs should be gone.

17. Cancel the other two jobs by indicating the user who sent them. Verify the result. For example:

```
# cancel -u root
# lpstat -o
```

18. Enable printing for your default printer.

```
# enable printername
```

19. Set your default printer to reject requests, and display a reason for doing so. For example:

```
# reject -r "Printer is down for maintenance" printername
```

20. Attempt to send a job to the default printer. Observe the messages displayed.

```
# lp /etc/hosts
```

Your message should say `printername: Requests are not being accepted`

21. Use the `lpstat` command to display the reason that the printer is not accepting requests. Use the following syntax:

```
# lpstat -a printername
```

Your message should say *`printername: your reason from step 20.`*

22. Set your default printer to again accept requests.

```
# accept printername
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 14

Controlling System Processes

Objectives

Upon completion of this module, you should be able to:

- View system processes
- Clear frozen processes
- Schedule an automatic one-time execution of a command
- Schedule an automatic recurring execution of a command

The following course map shows how this module fits into the current instructional goal.

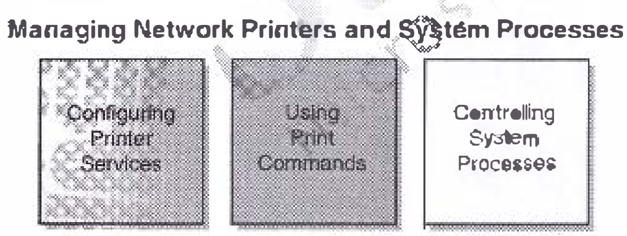

Figure 14-1 Course Map

Viewing System Processes

A process is any program that is running on the system. All processes are assigned a unique process identification (PID) number, which is used by the `kernel` to track and manage the process. The PID numbers are used by the `root` and regular users to identify and control their processes.

Using the CDE Process Manager

The Solaris OE Common Desktop Environment (CDE) provides a Process Manager to monitor and control processes that are running on the local system.

To start the Process Manager, click the Find Process control on the Tools subpanel of the Front Panel. Figure 14-2 shows the Tools menu.

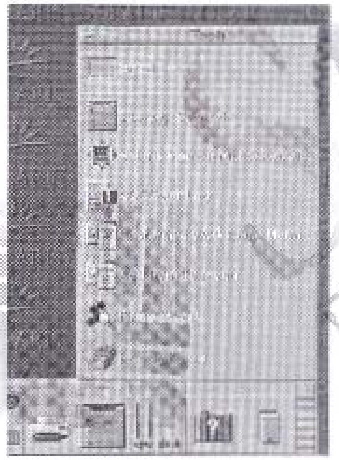

Figure 14-2 Tools Menu

You can also start the CDE Process Manager from the command line by typing the following:

```
# /usr/dt/bin/sdtProcess &
```

Figure 14-3 shows the window that appears.

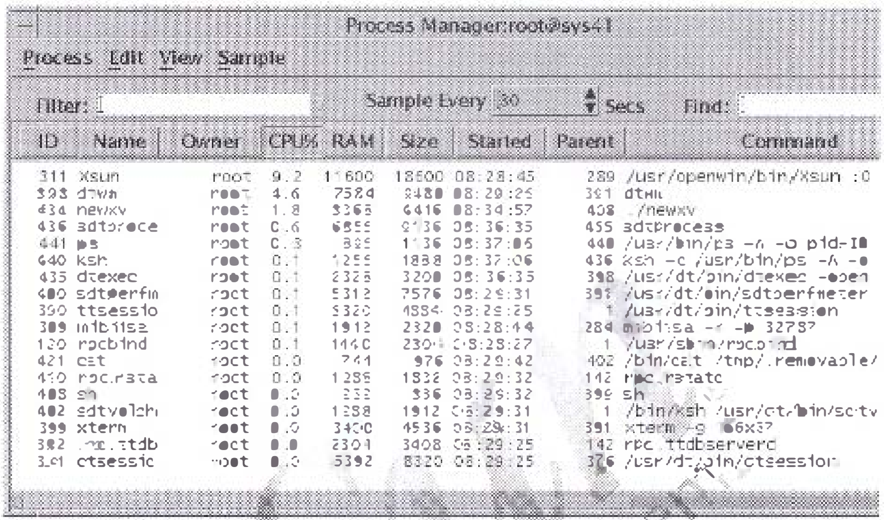

Figure 14-3 CDE Process Manager Window

The Process Manager can sort processes alphabetically (Name) or numerically (ID), depending on the column that is selected.

You can initiate a search by typing text into the Find field.

To terminate a process, highlight it and press Control-C, select the Kill option from the Process menu, or select the kill option from the options that are available when you press the right mouse button.

Using the `prstat` Command

The `prstat` command examines and displays information about active processes on the system.

This command enables you to view information by specific processes, user identification (UID) numbers, central processing unit (CPU) IDs, or processor sets. By default, the `prstat` command displays information about all processes sorted by CPU usage. To use the `prstat` command, perform the command:

```
# prstat
   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
  1257 root     4600K 4232K cpu0    39    0   0:00:00 0.3% prstat/1
  1249 root      328K  256K sleep   59    0   0:00:00 0.1% sh/1
  1247 root     1872K 1448K sleep   59    0   0:00:00 0.0% in.telnetd/1
  1256 root     1896K 1416K sleep   49    0   0:00:00 0.0% ksh/1
   243 root     2840K 2376K sleep   59    0   0:00:00 0.0% nscd/18
   388 root     2728K 1544K sleep   59    0   0:00:00 0.0% sshd/1
(output edited for brevity)
   209 root     3704K 2032K sleep   59    0   0:00:00 0.0% automountd/3
   228 root     2304K 1392K sleep   59    0   0:00:00 0.0% cron/1
    62 root     2848K 2112K sleep   59    0   0:00:00 0.0% picld/4
    55 root     2296K 1448K sleep   59    0   0:00:00 0.0% syseventd/13
   132 root     2184K 1368K sleep   59    0   0:00:00 0.0% rpcbind/1
Total: 48 processes, 206 lwps, load averages: 0.00, 0.00, 0.01
#
```

To quit the `prstat` command, type q.

Table 14-1 shows the column headings and their meanings in a `prstat` report.

Table 14-1 Column Headings for the `prstat` Report

Default Column Heading	Description
PID	The PID number of the process.
USERNAME	The login name or UID of the owner of the process.
SIZE	The total virtual memory size of the process.
RSS	The resident set size of the process in kilobytes, megabytes, or gigabytes.

Table 14-1 Column Headings for the `prstat` Report (Continued)

Default Column Heading	Description
`STATE`	The state of the process: • `cpu` – The process is running on the CPU. • `sleep` – The process is waiting for an event to complete. • `run` – The process is in the run queue. • `zombie` – The process terminated, and the parent is not waiting. • `stop` – The process is stopped.
`PRI`	The priority of the process.
`NICE`	The value used in priority computation.
`TIME`	The cumulative execution time for the process.
`CPU`	The percentage of recent CPU time used by the process.
`PROCESS/NLWP`	The name of the process/the number of lightweight processes (LWPs) in the process.

Note – The `kernel` and many applications are now multithreaded. A thread is a logical sequence of program instructions written to accomplish a particular task. Each application thread is independently scheduled to run on an LWP, which functions as a virtual CPU. LWPs in turn, are attached to `kernel` threads, which are scheduled to run on actual CPUs.

Note – Use the `priocntl(1)` command to assign processes to a priority class and to manage process priorities. The `nice(1)` command is only supported for backward compatibility to previous Solaris OE releases. The `priocntl` command provides more flexibility in managing processes.

Table 14-2 shows the options for the prstat command.

Table 14-2 Options for the prstat Command

Option	Description
-a	Displays separate reports about processes and users at the same time.
-c	Continuously prints new reports below previous reports.
-n *nproc*	Restricts the number of output lines.
-p *pidlist*	Reports only on processes that have a PID in the given list.
-s *key*	Sorts output lines by *key* in descending order. The five possible keys include: cpu, time, size, rss, and pri. You can use only one key at a time.
-S *key*	Sorts output lines by *key* in ascending order.
-t	Reports total usage summary for each user.
-u *euidlist*	Reports only processes that have an effective user ID (EUID) in the given list.
-U *uidlist*	Reports only processes that have a real UID in the given list.

Using the Solaris Management Console Process Tool

The Solaris Management Console provides a tool for monitoring and managing system processes. You open the Process Tool by clicking This Computer, and then clicking System Status. Then click Process.

Figure 14-4 shows the Solaris Management Console Process Tool.

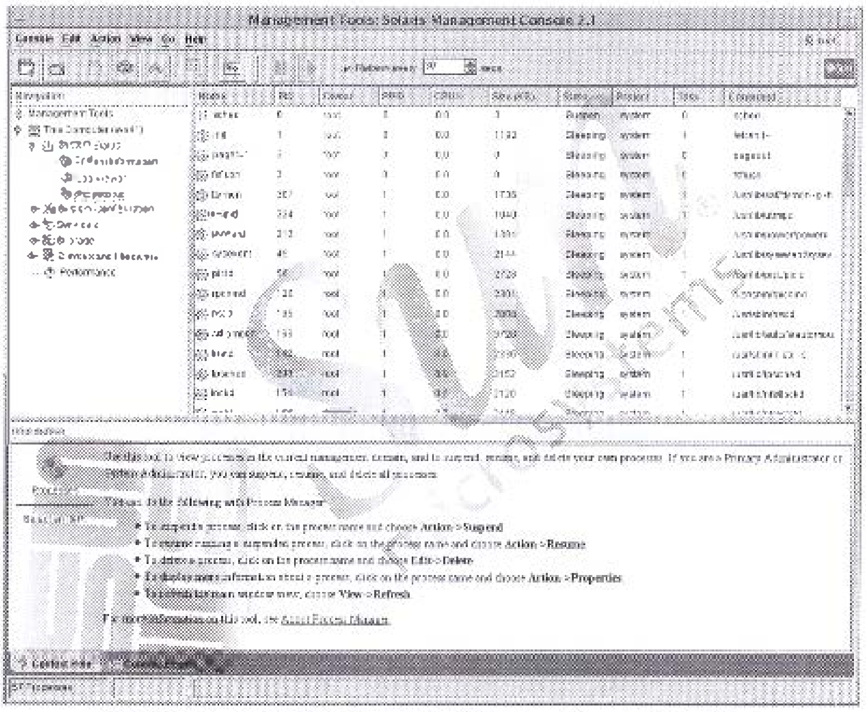

Figure 14-4 Solaris Management Console – Process Tool Window

From the Process Tool, you can do the following:

- Suspend a process. To do this, click the process name, and choose Suspend from the Action menu.
- Resume running a suspended process. To do this, click the process name, and choose Resume from the Action menu.
- Kill (delete) a process. To do this, click the process name, and choose Delete from the Edit menu.
- Display more information about a process. To do this, click the process name, and choose Properties from the Action menu.
- Refresh the main window view. To do this, choose Refresh from the View menu.

Clearing Frozen Processes

You use the kill command or the pkill command to send a signal to one or more running processes. You would typically use these commands to terminate or clear an unwanted process.

Using the kill and pkill Commands

You use the kill or pkill commands to terminate one or more processes.

The format for the kill command is:

```
kill -signal PID
```

The format for the pkill command is:

```
pkill -signal Process
```

Before you can terminate a process, you must know its name or PID. Use either the ps or pgrep command to locate the PID for the process.

The following examples uses the pgrep command to locate the PID for the mail processes.

```
$ pgrep -l mail
  551   sendmail
 12047 dtmail
$
$ pkill dtmail
```

The following examples use the ps and pkill commands to locate and terminate the dtmail process.

```
# ps -e |grep mail
   314 ?        0:00 sendmail
  1197 ?        0:01 dtmail
# kill 1197
```

To terminate more than one process at the same time, use the following syntax:

```
$ kill signal PID PID PID PID
$ pkill signal process process
```

You use the kill command without a signal on the command line to send the default Signal 15 to the process. This signal usually causes the process to terminate.

Table 14-3 shows some signals and names.

Table 14-3 Process Signal Numbers and Names

Signal Number	Signal Name	Event	Default Action
1	SIGHUP	Hangup	Exit
2	SIGINT	Interrupt	Exit
9	SIGKILL	Kill	Exit
15	SIGTERM	Terminate	Exit

- 1, SIGHUP – A hangup signal to cause a telephone line or terminal connection to be dropped. For certain daemons, such as inetd and in.named, a hangup signal will cause the daemon to reread its configuration file.
- 2, SIGINT – An interrupt signal from your keyboard—usually from a Control-C key combination.
- 9, SIGKILL – A signal to kill a process. A process cannot ignore this signal.
- 15, SIGTERM – A signal to terminate a process in an orderly manner. Some processes ignore this signal.

A complete list of signals that the kill command can send can be found by executing the command kill -l, or by referring to the man page for signal:

```
# man -s3head signal
```

Some processes can be written to ignore Signal 15. Processes that do not respond to a Signal 15 can be terminated by force by using Signal 9 with the kill or pkill commands. You use the following syntax:

```
$ kill -9 PID
$ pkill -9 process
```

Caution – Use the kill -9 or pkill -9 command as a last resort to terminate a process. Using the -9 signal on a process that controls a database application or a program that updates files can be disastrous. The process is terminated instantly with no opportunity to perform an orderly shutdown.

Performing a Remote Login

When a workstation is not responding to your keyboard or mouse input, the CDE might be frozen. In such cases, you may be able to remotely access your workstation by using the rlogin command or by using the telnet command from another system.

Killing the Process for a Frozen Login

After you are connected remotely to your system, you can invoke the pkill command to terminate the corrupted session on your workstation.

In the following examples, the rlogin command is used to log in to sys42, from which you can issue a pkill or a kill command.

```
$ rlogin sys42
Password: EnterPassword
Last login: Mon Jan 14 10:11:56 from sys43
Sun Microsystems Inc.   SunOS 5.9   Beta   May 2002
$ pkill -9 Xsun
or
$ ps -e | grep Xsun
379 ?        0:03 Xsun
$ kill -9 379
```

Scheduling an Automatic One-Time Execution of a Command

Use the at command to automatically execute a job only once at a specified time.

Using the at Command

The format for the at command is:

```
at -m -q queuename time date
at -r job
at -l
```

Table 14-4 shows the options you can use to instruct the cron process on how to execute an at job.

Table 14-4 Options for the at Command

Option	Description
-m	Sends mail to the user after the job has finished
-r job	Removes a scheduled at job from the queue
-q queuename	Specifies a specific queue
time	Specifies a time for the command to execute
-l	Reports all jobs scheduled for the invoking user
date	Specifies an optional date for the command to execute, which is either a month name followed by a day number or a day of the week

For example, to create an at job to run at 9:00 p.m. to locate and delete core files from user2's home directory, perform the command:

```
# at 9:00 pm
at> find /export/home/user2 -name core -exec rm {} \;
at> <Control-D>
commands will be executed using /sbin/sh
job 1016078400.a at Wed Mar 13 21:00:00 2002
```

To display information about the execution times of jobs, perform the command:

```
# at -l 1016078400.a
1016078400.a    Wed Mar 13 21:00:00 2002
```

To display the jobs queued to run at specified times by chronological order of execution, perform the command:

```
# atq
 Rank     Execution Date     Owner     Job           Queue   Job Name
  1st   Mar 13, 2002 21:00   root    1016078400.a      a       stdin
  2nd   Mar 13, 2002 21:05   root    1016078700.a      a       stdin
  3rd   Mar 13, 2002 21:10   root    1016079000.a      a       stdin
```

To view all the at jobs currently scheduled in the queue, perform the command:

```
# ls -l /var/spool/cron/atjobs
-r-Sr--r--   1 root     other        921 Mar 13 13:08 1016078400.a
-r-Sr--r--   1 root     other        915 Mar 13 13:08 1016078700.a
-r-Sr--r--   1 root     other        885 Mar 13 13:08 1016079000.a
```

You can also use the at command to remove a job from the at queue.

For example, to remove job 1016078400.a from the at queue, perform the command:

```
# at -r 1016078400.a
# atq
 Rank     Execution Date     Owner     Job           Queue   Job Name
  1st   Mar 13, 2002 21:05   root    1016078700.a      a       stdin
  2nd   Mar 13, 2002 21:10   root    1016079000.a      a       stdin
```

Controlling Access to the `at` Command

As the `root` user, you control who has access to the `at` command with the `at.deny` and `at.allow` files.

The `/etc/cron.d/at.deny` File

By default, the Solaris OE includes the `/etc/cron.d/at.deny` file. This file identifies users who are prohibited from using the `at` command. The file format is one user name per line. The file initially contains:

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

A user who is denied access to the `at` command receives the following message when attempting to use the command:

```
at: you are not authorized to use at. Sorry.
```

If only the `/etc/cron.d/at.deny` file exists but is empty, then all logged-in users can access the `at` command.

The `/etc/cron.d/at.allow` File

The `/etc/cron.d/at.allow` file does not exist by default, so all users (except those listed in the `/etc/cron.d/at.deny` file) can create `at` jobs. By creating the `/etc/cron.d/at.allow` file, you create a list of only those users who are allowed to execute `at` commands.

The `/etc/cron.d/at.allow` file consists of user names, one per line.

The interaction between the `at.allow` and the `at.deny` files follows these rules:

- If the `at.allow` file exists, only the users listed in this file can execute `at` commands.
- If the `at.allow` file does not exist, all users, except for users listed in the `at.deny` file, can execute `at` commands.
- If neither file exists, only the `root` user can use the `at` command.

Scheduling an Automatic Recurring Execution of a Command

You can use the `cron` facility to schedule regularly recurring commands. Users can submit a command to the `cron` facility by modifying their `crontab` file.

All `crontab` files are maintained in the `/var/spool/cron/crontabs` directory and are stored as the login name of the user that created the `cron` job.

The `cron` daemon is responsible for scheduling and running these jobs.

Note – The clock daemon, `cron`, starts at system boot and runs continuously in the background.

Introducing the `crontab` File Format

A `crontab` file consists of lines of six fields each. The fields are separated by spaces or tabs. The first five fields provide the date and time the command is to be scheduled. The last field is the full path to the command.

Note – If the command field contains a percent (%) character, then all subsequent characters are passed to the command as standard input.

These first five fields are separated by spaces and indicate when the command will be executed. See Figure 14-5.

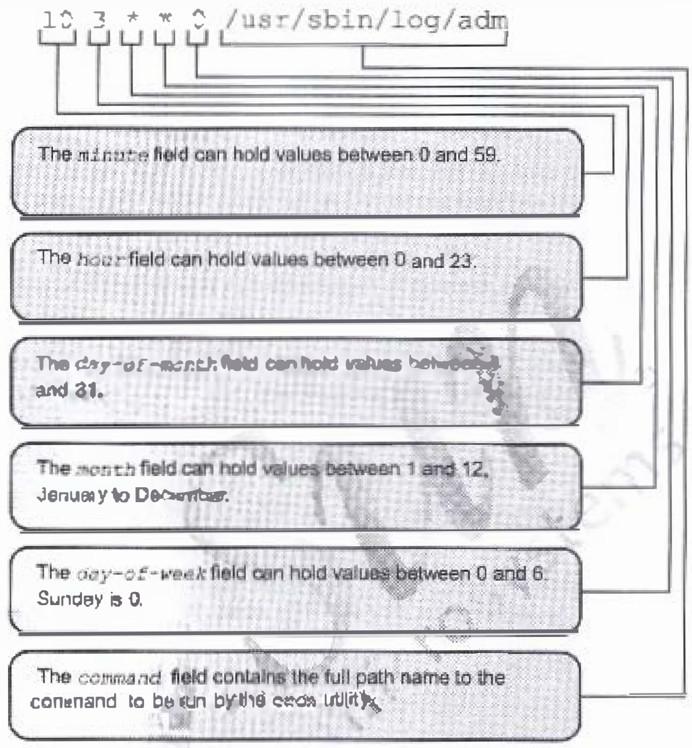

Figure 14-5 First Five Fields in a crontab File

The first five fields follow the format rules shown in Table 14-5.

Table 14-5 Rules for the crontab Fields

Value	Rule	Example
n	Matches if field value is *n*	As shown in the preceding figure for hour or minute, a 3 or 10
n,p,q	Matches if field value is *n*, *p*, or *q*	Every 10 minutes would be represented by 0,10,20,30,40,50
n-p	Matches if field has values between *n* and *p* inclusive	The hours between 1:00 a.m. and 4:00 a.m. would be represented by 1-4
*	Matches all legal values	As in the preceding example for the month, representing every month.

Using the crontab Command

The crontab command enables the user to view, edit, or remove a crontab file.

Viewing a crontab File

To view the contents of the root crontab file, run the crontab -l command as the root user.

```
# crontab -l
#ident  "@(#)root       1.20     01/11/06 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] &&
/usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

This is the same command that users run to view the contents of their own crontab file.

As the root user, you can view the contents of any regular user's crontab file by performing the command:

```
# crontab -l username
```

Editing a crontab File

To create or edit a crontab file, follow these steps:

1. Check that the EDITOR variable is set to the editor you want to use. This instructs the cron utility which editor to use to open the file.

```
# EDITOR=vi
# export EDITOR
```

2. Run the following crontab command to open your crontab file, and add the appropriate entry.

```
# crontab -e
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console
:wq
```

Note – If the users do not redirect the standard output and standard error of their commands in the crontab file, any generated output or errors are mailed electronically to the user.

Removing a crontab File

The correct way to remove a crontab file is to invoke the command:

```
# crontab -r username
```

Typical users can remove only their own crontab file. The root user can delete any user's crontab file.

Caution – If you accidently enter the crontab command on the command line without an option (-l, -e, -r), press the interrupt keys Control-C to exit. Do not press Control-D; this action overwrites the existing crontab file with an empty file.

Controlling Access to the `crontab` Command

You can control access to the `crontab` command with two files in the `/etc/cron.d` directory—the `cron.deny` file and the `cron.allow` file.

These files permit only specified users to perform `crontab` tasks, such as creating, editing, displaying, or removing their own `crontab` files.

The `/etc/cron.d/cron.deny` File

The Solaris OE provides a default `cron.deny` file. The file consists of a list of user names, one per line, of the users who are not allowed to use `cron`. The following is an example of the contents of a `cron.deny` file:

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

The `/etc/cron.d/cron.allow` File

The `/etc/cron.d/cron.allow` file does not exist by default, so all users (except those listed in the `cron.deny` file) can access their `crontab` file. By creating a `cron.allow` file, you can list only those users who can access `crontab` commands.

The file consists of a list of user names, one per line.

The interaction between the `cron.allow` and the `cron.deny` files follows these rules:

- If the `cron.allow` file exists, only the users listed in this file can create, edit, display, or remove `crontab` files.
- If the `cron.allow` file does not exist, all users, except for users listed in the `cron.deny` file, can create, edit, display, or remove `crontab` files.
- If neither file exists, only the `root` user can run the `crontab` command.

Using the Solaris™ Management Console Job Scheduler Tool

The Solaris™ Management Console contains a Scheduled Jobs tool to create and schedule jobs on your system. Users can manage jobs if the following conditions exist:

- Their user name appears in the /etc/cron.d/cron.allow file.
- Their user name does not appear in the /etc/cron.d/cron.deny file.
- The /etc/cron.d/cron.allow and /etc/cron.d/cron.deny files do not exist, and you are the root user.

To open the Job Scheduler from the Solaris Management Console, click This Computer and then click Service, and finally, click Scheduled Jobs.

See Figure 14-6 for an example of the Solaris Management Console Job Scheduler window.

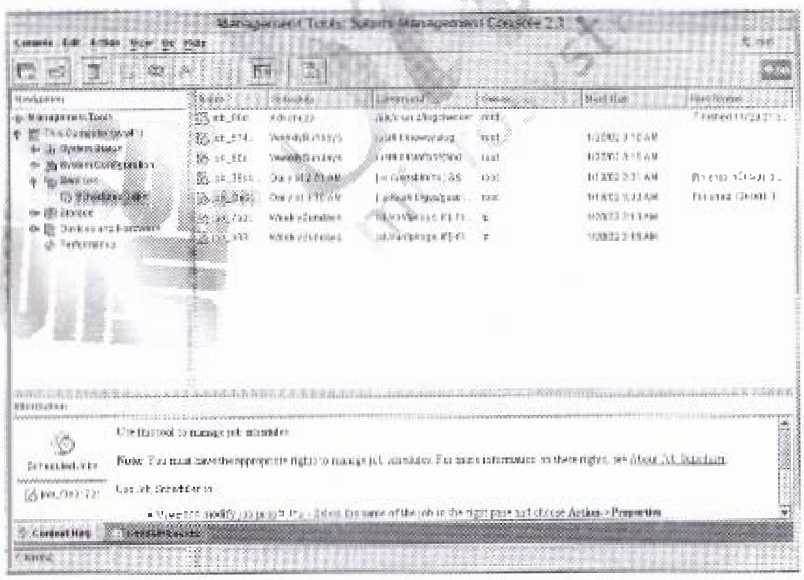

Figure 14-6 Solaris Management Console – Job Scheduler Window

You can use the Job Scheduler to:

- View and modify job properties

  Select the name of the job in the view pane, and choose Properties from the Action menu.

- Delete a job

  Select the job name and choose Delete from the Edit menu. The root user can delete all jobs. Users can only view and delete their own jobs.

- Add a scheduled job

  Choose Add Scheduled Job on the Action menu.

- Enable and disable job logging, and set search paths

  Choose Scheduled Job Policies on the Action menu.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using Process Control (Level 1)

In this exercise, you use the Process Tool and the prstat command to monitor and kill processes. You create an at job and create an entry in a crontab file.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Tasks

Complete the following tasks:

- Start the Process Tool. Run the prstat command in a window. In a separate window, run the find / command. Make note of the CPU percentages for the find command, as displayed by the prstat command and the Process Tool. Open a third window, and identify the PID of the shell running in it. Use the Process Tool to show the ancestry of the shell process. Use the Process Tool to kill the shell process. Use the Process Tool to send the TERM signal to the prstat process. Exit the Process Tool when you are finished.

  (Steps 1–6 in the Level 2 lab)
- Identify the device associated with your current terminal, and display the current time of day. Submit an at job that echoes Test Complete to your current window. Have the job run five minutes from the current time, and submit it to the queue called x. Display the at job in the queue.

  (Steps 7–10 in the Level 2 lab)
- Set the EDITOR variable to vi. Use the crontab command to determine when the logadm process is scheduled to run. Use the crontab command to edit the crontab file for the root user. Add an entry that sends the message It works! to your current window five minutes from the current time.

  (Steps 11–14 in the Level 2 lab)

Exercise: Using Process Control (Level 2)

In this exercise, you use the Process Tool and the prstat command to monitor and kill processes. You create an at job and create an entry in a crontab file.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Start the Process Tool. Run the prstat command in a window. In a separate window, run the find / command. Make note of the CPU percentages for the find command, as displayed by the prstat command and the Process Tool. Open a third window, and identify the PID of the shell running in it. Use the Process Tool to show the ancestry of the shell process. Use the Process Tool to kill the shell process. Use the Process Tool to send the TERM signal to the prstat process. Exit the Process Tool when you are finished.
- Identify the device associated with your current terminal, and display the current time of day. Submit an at job that echoes Test Complete to your current window. Have the job run five minutes from the current time, and submit it to the queue called x. Display the at job in the queue.
- Set the EDITOR variable to vi. Use the crontab command to determine when the logadm process is scheduled to run. Use the crontab command to edit the crontab file for the root user. Add an entry that sends the message It works! to your current window five minutes from the current time.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Start the Process Tool either by selecting the Find Process option from the Front Panel Tools menu in CDE or by invoking the appropriate command from the command line.

   In the Process Tool display, sort the listing according to `CPU%`, and change the sample time to five seconds.

2. Open a second terminal window, and run the `prstat` command.

3. Position the Process Tool and the window in which the `prstat` command is running so that you can observe both simultaneously. In an available window, run the `find` command to list all files on your system. Observe how the Process Tool and the `prstat` command display statistics for the `find` command.

   What is the maximum percentage of recent CPU time used by the `find` command as it executes?

4. Open a third terminal window, and run the `ps` command to determine the PID of the shell. Record the PID you find.

5. In the Process Tool, locate and select the shell process you identified in the previous step. Select the Show Ancestry option from the Process menu in the Process Tool. What is the name and PID of the first process listed?

6. Close the Show Ancestry window. Again, select the shell process you identified in Step 4. From the Process menu in the Process Tool, select the Kill option. What happens?

7. In the Process Tool, use the Find function to locate the `prstat` process. Select the Signal option from the Process menu. In the Signal fill-in field, enter the `TSTP` signal, and click OK. What happens to the `prstat` process? Close the Process Tool when you are finished.

8. Identify the device associated with your current terminal by using the `tty` command, and display the current time of day.

9. Submit an `at` job that echoes `Test Complete` to your current window. Have the job run five minutes from the current time, and submit it to the queue called x.

10. Display the `at` job in the queue.

11. Open a new window and set and export the EDITOR environment variable to use the vi editor to edit crontab files.

    If you are using the Bourne or Korn shell, perform the command:

```
# EDITOR=vi
# export EDITOR
```

    If you are using the C shell, perform the command:

```
# setenv EDITOR vi
```

12. Use the crontab command to view the current crontab file for the root user.
13. When is the logadm process scheduled to run?
14. Use the crontab command to edit the crontab file for the root user. Add an entry that sends the message It works! to your current window five minutes from now. For example, if the current time is 10:25, make an entry in your crontab file for the 30th minute of the same hour.

    Save the file, and quit the vi edit session. In about five minutes, you should see the result in your window.

Exercise: Using Process Control (Level 3)

In this exercise, you use the Process Tool and the `prstat` command to monitor and kill processes. You create an at job and create an entry in a `crontab` file.

Preparation

Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Start the Process Tool. Run the `prstat` command in a window. In a separate window, run the `find /` command. Make note of the CPU percentages for the `find` command, as displayed by the `prstat` command and the Process Tool. Open a third window, and identify the PID of the shell running in it. Use the Process Tool to show the ancestry of the shell process. Use the Process Tool to kill the shell process. Use the Process Tool to send the TERM signal to the `prstat` process. Exit the Process Tool when you are finished.
- Identify the device associated with your current terminal, and display the current time of day. Submit an at job that echoes `Test complete` to your current window. Have the job run five minutes from the current time, and submit it to the queue called x. Display the at job in the queue.
- Set the EDITOR variable to `vi`. Use the `crontab` command to determine when the `logadm` process is scheduled to run. Use the `crontab` command to edit the `crontab` file for the `root` user. Add an entry that sends the message `It works!` to your current window five minutes from the current time.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Start the Process Tool either by selecting the Find Process option from the Front Panel Tools menu in CDE or by invoking the appropriate command from the command line.

```
# /usr/dt/bin/sdtprocess &
```

   In the Process Tool display, sort the listing according to CPU%, and change the sample time to five seconds.

2. Open a second terminal window, and run the prstat command.

```
# prstat
```

3. Position the Process Tool and the window in which the prstat command is running so that you can observe both simultaneously. In an available window, run the find command to list all files on your system. Observe how the Process Tool and the prstat command display statistics for the find command.

```
# find /
```

   What is the maximum percentage of recent CPU time used by the find command as it executes?

   This varies according to your system configuration. Some systems might display values in the 20-percent range.

4. Open a third terminal window, and run the ps command to determine the PID of the shell. Record the PID you find.

```
# ps
```

   Your value appears here.

5. In the Process Tool, locate and select the shell process you identified in the previous step. Select the Show Ancestry option from the Process menu in the Process Tool. What is the name and PID of the first process listed?

   The PID varies. On systems running the CDE, the first process listed should be /usr/dt/bin/dtlogin.

6. Close the Show Ancestry window. Again, select the shell process you identified in Step 4. From the Process menu in the Process Tool, select the Kill option. What happens?

   The process stops, and the window no longer appears.

7. In the Process Tool, use the Find function to locate the prstat process. Select the Signal option from the Process menu. In the Signal fill-in field, enter the TERM signal, and click OK. What happens to the prstat process? Close the Process Tool when you are finished.

   The prstat process terminates, and the prompt appears in the window in which it ran.

8. Identify the device associated with your current terminal by using the tty command, and display the current time of day.

```
# tty
(something like /dev/pts/4 should appear
# date
(current date/time appears)
```

9. Submit an at job that echoes Test Complete to your current window. Have the job run five minutes from the current time, and submit it to the queue called x.

```
# at -q x 13:30
at> echo "Test Complete" > /dev/pts/# (# is from the tty command)
at> <Control-D>
commands will be executed using /sbin/sh
job 958163400.x at Fri May 12 13:30:00 2000
#
```

10. Display the at job in the queue.

```
# atq
```

11. Open a new window and set and export the EDITOR environment variable to use the vi editor to edit crontab files.

    If you are using the Bourne or Korn shell, perform the command:

```
# EDITOR=vi
# export EDITOR
```

    If you are using the C shell, perform the command:

```
# setenv EDITOR vi
```

12. Use the crontab command to view the current crontab file for the root user.

```
# crontab -l
```

13. When is the logadm process scheduled to run?

    Ten minutes after 3:00 a.m. on all days

14. Use the crontab command to edit the crontab file for the root user. Add an entry that sends the message It works! to your current window five minutes from now. For example, if the current time is 10:25, make an entry in your crontab file for the 30th minute of the same hour.

```
# tty
/dev/pts/#
# date
Thu May 11 10:25:14 PDT 2000
# crontab -e
```

Add the following line, but substitute the correct time and terminal device:

```
30  10 * * * /usr/bin/echo "It works!" > /dev/pts/#
```

Save the file, and quit the vi edit session. In about five minutes, you should see the result in your window.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Performing File System Backups

Objectives

Upon completion of this module, you should be able to:

- Identify the fundamentals of backups
- Back up an unmounted file system

The following course map shows how this module fits into the current instructional goal.

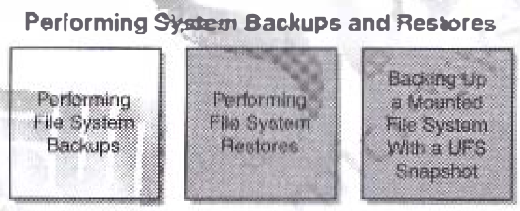

Figure 15-1 Course Map

Introducing the Fundamentals of Backups

A crucial function of system administration is to backup file systems. Backups safeguard against data loss, damage, or corruption. Backup tapes are often referred to as dump tapes.

Importance of Routine File System Backups

To back up file systems, you copy file systems to removable media, such as a tape. You perform backups on a regular basis to prevent loss of data due to:

- Accidental deletion of files
- Hardware failures
- Problems with re-installations or system upgrades
- System crashes
- System break-ins by unauthorized users, compromising data integrity
- Natural disasters

Tape Media Types

Table 15-1 shows typical tape media that you can use to store file systems during the backup process. Select media based on the availability of equipment and your preference.

Table 15-1 Tape Media Types

Media Type	Capacity
1/2-inch reel tape	140 Mbytes (6250 bits per inch)
1/4-inch cartridge (QIC)[1] cartridge tape	8 Gbytes
8-mm cartridge tape	40 Gbytes
4-mm digital audio tape (DAT)[2] cartridge tape	24 Gbytes
DLT[3] 1/2-inch cartridge tape	70 Gbytes
LTO[4] cartridge tape	100 Gbytes

1. QIC stands for quarter-inch tape. 2. DAT stands for digital audio tape.
3. DLT stands for digital linear tape. 4. LTO stands for linear tape open.

The capacities in the table are approximate. Tape capacity increases with new technology. Check the documentation that comes with the tape device to determine the capacity.

Tape Drive Naming

All tape drives have logical device names that you use to reference the device on the command line. Figure 15-2 shows the format that all logical device names use.

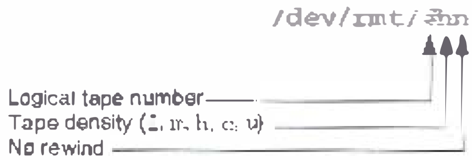

Figure 15-2 Logical Device Name Format

The logical tape numbers in the tape drive names always start with 0. For example:

- The first instance of a tape drive:

  /dev/rmt/0
- The second instance of a tape drive:

  /dev/rmt/1
- The third instance of a tape drive:

  /dev/rmt/2

Two optional parameters further define the logical device name:

- Tape density – Five values can be given in the tape device name: l (low), m (medium), h (high), c (compressed), or u (ultra compressed).
- No rewind – The letter n at the end of a tape device name indicates that the tape should not be rewound when the current operation completes.

Tape densities depend on the tape drive. Check the manufacturer's documentation to determine the correct densities for the tape media.

Tape drives that support data compression contain internal hardware that performs the compression. Hardware compression uses more space than the software compression you can achieve from the Solaris™ Operating Environment (Solaris OE) compress command, but compression is muc faster. If you back up a software-compressed file with hardware compression, the resultant file is larger in size.

Tape Drive Control

You use the mt command (magnetic tape control) to send instructions to the tape drive. Not all tape drives support all mt commands.

The format for the mt command is:

```
mt -f tape-device-name command count
```

You use the -f option to specify the tape device name, typically a no-rewind device name.

Using the mt Command

Table 15-2 lists some of the mt commands that you can use to control a magnetic tape drive.

Table 15-2 Definitions of mt Commands

Command	Definition
mt status	Displays status information about the tape drive
mt rewind	Rewinds the tape
mt offline	Rewinds the tape and, if appropriate, takes the drive unit offline and if the hardware supports it, unloads
mt fsf *count*	Moves the tape forward *count* records

The following command positions the tape at the beginning of the third tape record.

```
# mt -f /dev/rmt/0n fsf 2
```

Strategies for Scheduled Backups

The most common method to schedule backups is to perform cumulative incremental backups daily. This schedule is recommended for most situations.

To set up a backup schedule, determine:

- The file systems to back up
- A backup device (for example, tape drive)
- The number of tapes to use for the backup
- The type of backup (for example, full or incremental)
- The procedures for marking and storing tapes
- The time it takes to perform a backup

Determining File System Names to Back Up

Display the contents of the `/etc/vfstab` file. Then view the `mount point` column to find the name of the file system that you want to back up.

Determining the Number of Tapes

You determine the number of tapes for a backup according to the size of the file system you are backing up.

To determine the size of the file system, use the `ufsdump` command with the `S` option. The following are the command formats:

```
# ufsdump 0S filesystem_name
<number reported>
```

or

```
# ufsdump 3S filesystem_name
<number reported>
```

The numeric option determines the appropriate dump level. The output is the estimated number of bytes that the system requires for a complete backup.

Divide the reported bytes by the capacity of the tape to determine how many tapes you need to backup the file system.

Determining Back Up Frequency and Levels

You determine how often and at what level to backup each file system. The level of a backup refers to the amount of information that is backed up.

Identifying Incremental and Full Back Ups

You can perform a full backup or an incremental backup of a file system. A full backup is a complete file system backup. An incremental backup copies only files in the file system that have been added or modified since a previous lower-level backup.

You use Dump Level 0 to perform a full backup. You use Dump Levels 1 through 9 to schedule incremental backups. The level numbers have no meaning other than their relationship to each other as a higher or lower number.

Figure 15-3 shows an example of a file system backup performed in incremental levels.

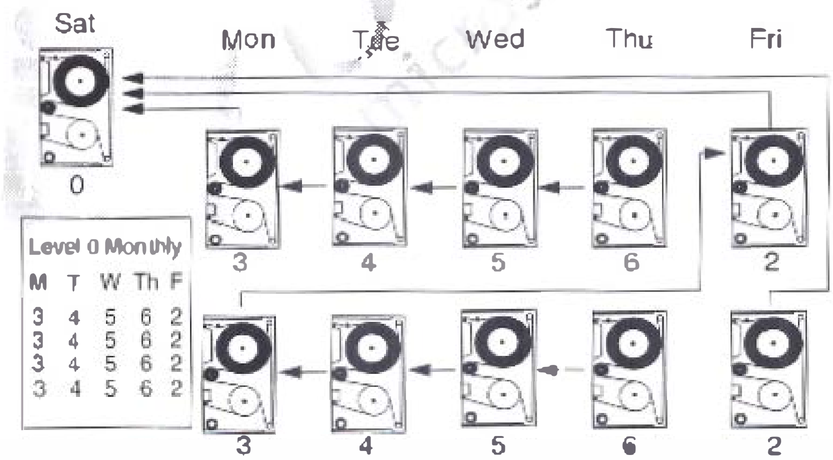

Figure 15-3 Incremental Back Up Strategy

Table 15-3 defines the elements of the incremental backup strategy shown in Figure 15-3.

Table 15-3 Incremental Back Up Level Definitions

Level	Example
0 (Full)	Performed once each month.
3	Performed every Monday. The backup copies new or modified files since the last lower-level backup (for example, 0).
4	Performed every Tuesday. The backup copies new or modified files since the last lower-level backup (for example, 3).
5	Performed every Wednesday. The backup copies new or modified files since the last lower-level backup (for example, 4).
6	Performed every Thursday. The backup copies new or modified files since the last lower-level backup (for example, 5).
2	Performed every Friday. The backup copies new or modified files since the last lower-level backup, which is the Level 0 backup at the beginning of the month.

Note – Many system administrators use the `crontab` utility to start a script that runs the `ufsdump` command.

The /etc/dumpdates File

The /etc/dumpdates file records backups if the -u option is used with the ufsdump command. Each line in the /etc/dumpdates file shows the file system that was backed up and the level of the last backup. It also shows the day, the date, and the time of the backup.

The following is an example /etc/dumpdates file.

```
# cat /etc/dumpdates
/dev/rdsk/c0t2d0s6  0 Fri Jan 4   19:12:27  2002
/dev/rdsk/c0t2d0s0  0 Fri Jan 4   20:44:02  2002
/dev/rdsk/c0t0d0s7   0 Tue Mar 12  09:58:26  2002
/dev/rdsk/c0t0d0s7   1 Tue Mar 12  16:25:28  2002
```

When an incremental backup is performed, the ufsdump command consults the /etc/dumpdates file. It looks for the date of the next lower-level backup. Then, the ufsdump command copies to the backup media all of the files that were modified or added since the date of that lower-level backup.

When the backup is complete, the /etc/dumpdates file records a new entry that describes this backup. The new entry replaces the entry for the previous backup at that level.

You can view the /etc/dumpdates file to determine if the system is completing backups. If a backup does not complete because of equipment failure, the /etc/dumpdates file does not record the backup.

Note – When you are restoring an entire file system, check the /etc/dumpdates file for a list of the most recent dates and levels of backups. Use this list to determine which tapes are needed to restore the entire file system. The tapes should be physically marked with the dump level and date of the backup.

Backing Up an Unmounted File System

Check that the file system is inactive, or unmounted, before you back the system up. If the file system is active, the output of the backup can be inconsistent, and you could find it impossible to restore the files correctly.

The ufsdump Command

The standard Solaris OE command for ufs file system backups is /usr/sbin/ufsdump.

The format for the ufsdump command is:

```
ufsdump option(s) argument(s) filesystem_name
```

You can use this command to back up a complete or a partial file system. Backups are often referred to as dumps.

Options for the ufsdump Command

Table 15-4 defines several common options for the ufsdump command.

Table 15-4 Options for the ufsdump Command

Option	Description
0–9	Back up level. Level 0 is a full backup of the file system. Levels 1 through 9 are incremental backups of files that have changed since the last lower-level backup.
v	Verify. After each tape is written, the system verifies the content of the media against the source file system. If any discrepancies occur, the system prompts the operator to insert new media and repeat the process. Use this option only on an unmounted file system. Any activity in the file system causes the system to report discrepancies.
S	Size estimate. This option allows you to estimate the amount of space that will be needed on the tape to perform the level of backup you want.
l	Autoload. You use this option with an autoloading (stackloader) tape drive.
o	Offline. When the backup is complete, the system takes the drive offline, rewinds the tape (if you use a tape), and, if possible, ejects the media.
u	Update. The system creates an entry in the /etc/dumpdates file with the device name for the file system disk slice, the backup level (0–9), and the date. If an entry already exists for a backup at the same level, the system replaces the entry.
n	Notify. The system sends messages to the terminals of all logged-in users who are members of the sys group to indicate that the ufsdump command requires attention.
f *device*	Specify. The system specifies the device name of the file system backup. When you use the default tape device, /dev/rmt/0, you do not need the -f option. The system assumes the default.

Tape Back Ups

You use the ufsdump command to create file system backups to tape. The dump level (0–9) specified in the ufsdump command determines which files to back up.

Using the ufsdump Command

Perform the following steps to use the ufsdump command to start a tape backup:

1. Become the root user to change the system to single-user mode, and unmount the file systems.

```
# /usr/sbin/shutdown -y -g300 "System is being shutdown for backup"

Shutdown started.    Thu 24 Jan 2002 01:01:47 PM MST

Broadcast Message from root (pts/3) on host1 Thu Jan 24 01:01:52 ...
The system host1 will be shut down in 5 minutes
System is being shutdown for backup
```

2. Verify that the /export/home file system was unmounted with the shutdown command. If not, unmount it manually.
3. Check the integrity of the file system data with the fsck command.

```
# fsck /export/home
```

4. Perform a full (Level 0) backup of the /export/home file system.

```
# ufsdump 0uf /dev/rmt/0 /export/home
ufsdump 0uf /dev/rmt/0 /export/home
  DUMP: Writing 32 Kilobyte records
  DUMP: Date of this level 0 dump: Thu 24 Jan 2002 01:06:47 PM MST
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (host1:/export/home) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 1126 blocks (563KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 1086 blocks (543KB) on 1 volume at 1803 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Thu 24 Jan 2002 01:06:47 PM MST
#
```

Remote Backups to a Tape

You can use the ufsdump command to perform a backup on a remote tape device.

The format for the ufsdump command is:

```
ufsdump options remotehost:tapedevice filesystem
```

To perform remote backups across the network, the system with the tape drive must have an entry in its /.rhosts file for every system that uses the tape drive.

Using the ufsdump Command

The following example shows how to perform a full (Level 0) backup of the /export/home file system on the host1 system, to the remote tape device on the host2 system.

```
# ufsdump 0uf host2:/dev/rmt/0 /export/home
  DUMP: Writing 32 Kilobyte records
  DUMP: Date of this level 0 dump: Thu 24 Jan 2002 01:13:55 PM MST
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (host1:/export/home) to
host2:/dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 320 blocks (160KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 318 blocks (159KB) on 1 volume at 691 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Thu 24 Jan 2002 01:13:55 PM MST
#
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Backing Up a File System (Level 1)

In this exercise, you back up an available file system on your system.

Preparation

This exercise requires a system that is configured with a tape drive and a file system that is available to unmount. This exercise assumes that the /export/home file system exists on a separate partition from the / (root) file system and can be unmounted. Identify the slice on which the /export/home file system resides. Get a tape that is appropriate for your system from the instructor.

Tasks

Complete the following tasks:

- Use the mt command to rewind the tape to the beginning.
- Use the ufsdump command to create a tape backup of the /export/home file system. Make sure that the /etc/dumpdates file is updated.

  (Steps 1–4 in the Level 2 lab)
- Add files and directories to the /export/home file system.

  (Steps 5–6 in the Level 2 lab)
- Use the ufsdump command to do an incremental backup of the /export/home file system.

  (Steps 7–9 in the Level 2 lab)
- Use the mt command to remove the tape from the tape drive.
- Review the /etc/dumpdates file.

  (Steps 10–12 in the Level 2 lab)

Exercise: Backing Up a File System (Level 2)

In this exercise, you back up an available file system on your system.

Preparation

This exercise requires a system that is configured with a tape drive and a file system that is available to unmount. This exercise assumes that the `/export/home` file system exists on a separate partition from the / (root) file system and can be unmounted. Identify the slice on which the `/export/home` file system resides. Get a tape that is appropriate for your system from the instructor.

Task Summary

In this exercise, you accomplish the following:

- Use the `mt` command to rewind the tape to the beginning.
- Use the `ufsdump` command to create a tape backup of the `/export/home` file system.
- Add files and directories to the `/export/home` file system.
- Use the `ufsdump` command to do an incremental backup of the `/export/home` file system.
- Use the `mt` command to remove the tape from the tape drive.
- Review the `/etc/dumpdates` file.

Tasks

Complete the following steps:

1. Unmount the `/export/home` file system. If your system reports that the `/export/home` file system is busy, use the `umount -f` command.
2. Insert a tape into your tape drive.
3. Use the `mt` command to rewind the tape to the beginning.
4. Use the `ufsdump` command to create a backup for the `/export/home` file system. Make sure that the `/etc/dumpdates` file is updated.
5. Mount the `/export/home` file system.

6. Copy the contents of the `/etc/uucp` directory to the `/export/home` directory.
7. Unmount the `/export/home` file system.
8. Move the tape to the next tape record.
9. Use the `ufsdump` command to create an incremental backup for the `/export/home` file system, using a non-rewinding device.
10. Rewind and eject the tape from the tape drive.
11. Set the tape aside for use with subsequent labs.
12. Review the contents of the `/etc/dumpdates` file.
13. Mount the `/export/home` file system.

Exercise: Backing Up a File System (Level 3)

In this exercise, you back up an available file system on your system.

Preparation

This exercise requires a system that is configured with a tape drive and file system that is available to unmount. This exercise assumes that the /export/home file system exists on a separate partition from the / (root) file system and can be unmounted. Identify the slice on which the /export/home file system resides. Get a tape that is appropriate for your system from the instructor.

Task Summary

In this exercise, you accomplish the following:

- Use the mt command to rewind the tape to the beginning.
- Use the ufsdump command to create a tape backup of the /export/home file system.
- Add files and directories to the /export/home file system.
- Use the ufsdump command to do an incremental backup of the /export/home file system.
- Use the mt command to remove the tape from the tape drive.
- Review the /etc/dumpdates file.

Tasks and Solutions

Complete the following steps:

1. Unmount the /export/home file system. If your system reports that the /export/home file system is busy, use the umount -f command.

```
# umount /export/home
```

2. Insert a tape into your tape drive.
3. Use the mt command to rewind the tape to the beginning.

```
# mt rewind
```

4. Use the ufsdump command to create a backup tape for the /export/home file system, where c#t#d#s# represents. If you cannot remember which device the /export/home file system was mounted on, view the contents of the /etc/vfstab file with the more command.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c#t#d#s#
```

You should see output similar to:

```
ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0d0t0s7
  DUMP: Writing 32 Kilobyte records
  DUMP: Date of this level 0 dump: Thu 24 Jan 2002 01:06:47 PM MST
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (sys43:/export/home) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 1125 blocks (563KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 1086 blocks (543KB) on 1 volume at 1803 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Thu 24 Jan 2002 01:06:47 PM MST
```

5. Mount the /export/home file system.

```
# mount /export/home
```

6. Copy the contents of the /etc/uucp directory to the /export/home directory.

```
# cp -r /etc/uucp /export/home
```

7. Unmount the /export/home file system.

```
# umount /export/home
```

8. Move the tape to the next tape record.

```
# mt -f /dev/rmt/0n fsf 1
```

9. Use the ufsdump command to create an incremental backup for the /export/home file system, using a non-rewinding device.

```
# ufsdump 1uf /dev/rmt/0n /dev/rdsk/c#t#d#s#
```

You should see output similar to:

```
ufsdump 1uf /dev/rmt/0n /dev/rdsk/c0d0t0s7
DUMP: Writing 32 Kilobyte records
  DUMP: Date of this level 0 dump: Thu 24 Jan 2002 01:13:55 PM MST
  DUMP: Date of last level 0 dump: Thu 24 Jan 2002 01:06:47 PM MST
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (sys43:/export/home) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 320 blocks (160KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 318 blocks (159KB) on 1 volume at 691 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 1 dump on Thu 24 Jan 2002 01:13:55 PM MST
```

10. Rewind and eject the tape from the tape drive.

```
# mt offline
```

11. Set the tape aside for use with subsequent labs.

12. Review the contents of the /etc/dumpdates file.

```
# more /etc/dumpdates
```

You should see one line showing information for the Level 0 dump and another line for the Level 1 dump, for example:

```
/dev/rdsk/c0t0d0s7          0 Thu Jan 24 13:20:49 2002
/dev/rdsk/c0t0d0s7          1 Thu Jan 24 13:22:06 2002
```

13. Mount the /export/home file system.

```
# mount /export/home
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 16

Performing File System Restores

Objectives

Upon completion of this module, you should be able to:

- Restore ufs file systems
- Explain disaster recovery fundamentals

The following course map shows how this module fits into the current instructional goal.

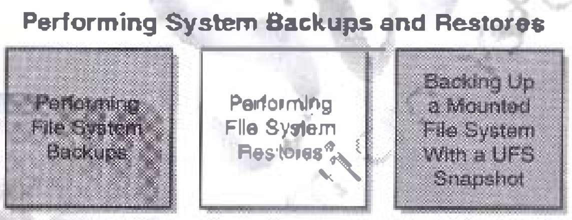

Figure 16-1 Course Map

Restoring a `ufs` File System

You restore a file system to rebuild a damaged file system, to reinstall or upgrade the Solaris OE software, or to reorganize file systems on existing or new disks.

Restoring a Regular File System

When you are restoring data to a system, consider the following questions:

- Can the system boot on its own (regular file system restore)?
- Do you need to boot the system from CD-ROM (critical file system restore)?
- Do you need to boot the system from CD-ROM and repair the boot drive (special case recovery)?

To restore files or file systems, determine the following:

- The file system backup tapes that are needed
- The device name to which you will restore the file system
- The name of the temporary directory to which you will restore individual files
- The type of backup device to be used (local or remote)
- The backup device name (local or remote)

To restore a regular file system, such as the `/export/home` or `/opt` file system, back up to the disk, you use the `ufsrestore` command. The `ufsrestore` command copies files to the disk, relative to the current working directory, from backup tapes that were created by the `ufsdump` command.

You can use the `ufsrestore` command to reload an entire file system hierarchy from a Level 0 backup and related incremental backups. You can also restore one or more single files from any backup tape.

The format for the `ufsrestore` command is:

```
ufsrestore option(s) argument(s) filesystem
```

Table 16-1 describes some options that you can use with the ufsrestore command.

Table 16-1 Options for the ufsrestore Command

Option	Description
t	Lists the table of contents of the backup media.
r	Restores the entire file system from the backup media.
x *file1 file2*	Restores only the files named on the command line.
i	Invokes an interactive restore.
v	Specifies verbose mode. This mode displays the path names to the terminal screen as each file is restored.
f *device*	Specifies the tape device name.

When you restore an entire file system from a backup tape, the system creates a restoresymtable file. The ufsrestore command uses the restoresymtable file for check-pointing or passing information between incremental restores. You can remove the restoresymtable file when the restore is complete.

Using the ufsrestore Command to Restore the /opt File System

The following procedure demonstrates how to use the ufsrestore command to restore the /opt file system on the c0t0d0s5 slice.

1. Create the new file system structure.

```
# newfs /dev/rdsk/c0t0d0s5
```

2. Mount the file system to the /opt directory, and change to that directory.

```
# mount /dev/dsk/c0t0d0s5 /opt
# cd /opt
```

3. Restore the /opt file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Always restore a file system by starting with the Level 0 backup tape, continuing with the next-lower-level tape, and continuing through the highest-level tape.

4. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

5. Unmount the new file system.

```
# cd /
# umount /opt
```

6. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s5
```

7. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s5
```

Note – Always back up the newly created file system because the `ufsrestore` command repositions the files and changes the inode allocation.

Restoring the /usr File System

To restore the `/usr` file system, boot from the Solaris 9 Software 1 of 2 CD-ROM, and then use the `ufsrestore` command to restore files back to the `/usr` partition.

Note – If the / (root), `/usr`, or `/var` file systems are unusable because of some type of corruption or damage, the system will not boot.

Using the `ufsrestore` Command to Restore the `/usr` File System

The following procedure demonstrates how to restore the `/usr` file system on Slice 6 of the boot disk.

1. Insert the Solaris 9 Software 1 of 2 CD-ROM, and boot from the CD-ROM with the single-user mode option.

```
ok boot cdrom -s
```

2. Create the new file system structure.

```
# newfs /dev/rdsk/c0t0d0s6
```

3. Mount the file system to the mount point /a, and change to that directory.

```
# mount /dev/dsk/c0t0d0s6 /a
# cd /a
```

4. Restore the /usr file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Remember to restore a file system by starting with the Level 0 backup tape, continuing with the next-lower-level tape, and continuing through the highest-level tape.

5. Remove the `restoresymtable` file.

```
# rm restoresymtable
```

6. Unmount the new file system.

```
# cd /
# umount /a
```

7. Use the `fsck` command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s6
```

8. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s6
```

9. Reboot the system.

```
# init 6
```

Performing a Special Case Recovery of the / (root) File System

You perform a special case recovery to recover the / (root) file system if there is damage to the boot block.

To restore the / (root) file system, boot from the Solaris 9 Software 1 of 2 CD-ROM, and use the ufsrestore command.

The following procedure demonstrates how to restore the / (root) file system on Slice 0 of the boot disk.

1. Insert the Solaris 9 Software 1 of 2 CD-ROM, and boot the CD-ROM with the single-user mode option.

```
ok boot cdrom -s
```

2. Create the new file system structure.

```
# newfs /dev/rdsk/c0t0d0s0
```

3. Mount the file system to the mount point /a and change to that directory.

```
# mount /dev/dsk/c0t0d0s0 /a
# cd /a
```

4. Restore the / (root) file system from the backup tape.

```
# ufsrestore rf /dev/rmt/0
```

Note – Always restore a file system by starting with the Level 0 backup tape, and continuing with the next-lower-level tape, and continuing through the highest-level tape.

---

5. Remove the restoresymtable file.

```
# rm restoresymtable
```

6. Install the bootblk in Sectors 1 through 15 of the boot disk. To do this, change to the directory that contains the bootblk, and enter the installboot command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
# installboot bootblk /dev/rdsk/c0t0d0s0
```

7. Unmount the new file system.

```
# cd /
# umount /a
```

8. Use the fsck command to check the restored file system.

```
# fsck /dev/rdsk/c0t0d0s0
```

9. Perform a full backup of the file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s0
```

10. Reboot the system.

```
# init 6
```

Invoking an Interactive Restore

The ufsrestore i command invokes an interactive interface. Through the interface, you can browse the directory hierarchy of the back up tape and select individual files to extract.

Using the ufsrestore i Command

The following procedure demonstrates how to use the ufsrestore i command to extract individual files from a backup tape.

1. Become the root user, and change to the temporary directory that you want to receive the extracted files.

```
# cd /var/tmp
```

2. Perform the ufsrestore i command.

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Fri Jan 25 [illegible]:38:53 2002
Dumped from: the epoch
Level 0 dump of /export/home on sys43:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
```

3. Display the contents of the directory structure on the backup tape.

```
ufsrestore > ls
.:
     2 *./          13 directory1     15 directory3     11 file2
     2 *../         14 directory2     10 file1          12 file3
```

4. Change to the target directory on the backup tape.

```
ufsrestore > cd directory1
ufsrestore > ls
./directory1:
  3904  ./          2 *../      3905  file1     3906  file2     3907  file3
```

5. Add the files you want to restore to the extraction list.

```
ufsrestore > add file1 file2
Make node ./directory1
```

Files you want to restore are marked with an asterisk (*) for extraction. If you extract a directory, all of the directory contents are marked for extraction.

In this example, two files are marked for extraction. The `ls` command displays an asterisk in front of the selected file names, `file1` and `file2`.

```
ufsrestore > ls
./directory1:
  3904 *./          2 *../      3905 *file1     3906 *file2     3907  file3
```

6. To delete a file from the extraction list, use the `delete` command.

```
ufsrestore > delete file1
```

The `ls` command displays the `file1` file without an asterisk.

```
ufsrestore > ls
./directory1:
  3904 *./          2 *../      3905  file1     3906 *file2     3907  file3
```

7. To view the files and directories marked for extraction, use the `marked` command.

```
ufsrestore > marked
./directory1:
  3904 *./          2 *../      3906 *file2
```

8. To restore the selected files from the backup tape, perform the command:

```
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
```

Note – The ufsrestore command has to find the selected files. If you used more than one tape for the backup, first insert the tape with the highest volume number and type the appropriate number at this point. Then repeat, working towards Volume #1 until all files have been restored.

```
extract file ./directory1/file2
Add Links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
```

Note – Answering y sets ownership and permissions of the temporary directory to those of the directory structure on the tape.

9. To exit the interactive restore after the files are extracted, perform the command:

```
ufsrestore> quit
```

10. Move the restored files to their original or permanent directory location, and delete the files from the temporary directory.

```
# mv /var/tmp/directory1/file2 /export/home
# rm -r /var/tmp/directory1
```

Note – You can use the help command in an interactive restore to display a list of available commands.

Performing an Incremental Restore

When performing incremental restores, start with the last volume and work towards the first. The system uses information in the restoresymtable file to restore incremental backups on top of the latest full backup.

Note – If you perform an incremental restore of data from backup tapes that were written from an active file system, the ufsrestore command might become disrupted.

The following procedure demonstrates how to restore the `/export/home` file system from incremental tapes.

Note – This procedure makes use of the interactive restore to assist in showing the concept of incremental restores. You would typically use a command, such as `ufsrestore rf`, for restoring entire file systems.

---

1. View the contents of the `/etc/dumpdates` file for information about the `/export/home` file system.

```
# more /etc/dumpdates |grep c0t0d0s7
/dev/rdsk/c0t0d0s7          0 Mon Jan 28 13:10:22 2002
/dev/rdsk/c0t0d0s7          1 Mon Jan 28 13:12:41 2002
```

2. Create the new file system structure for the `/export/home` file system.

```
# newfs /dev/rdsk/c0t0d0s7
```

3. Mount the file system and change to that directory.

```
# mount /dev/dsk/c0t0d0s7 /export/home
# cd /export/home
```

4. Insert the Level 0 backup tape.

5. Restore the `/export/home` file system from the backup tapes.

```
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Wed Apr 03 09:55:34 2002
Dumped from: the epoch
Level 0 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Make node ./directory1
Make node ./directory2
Make node ./directory3
Extract new leaves.
Check pointing the restore
extract file ./file1
extract file ./file2
extract file ./file3
Add links
Set directory mode, owner, and times.
Check the symbol table.
```

```
Check pointing the restore
#
```

6. Load the next lower-level tape into the tape drive.

```
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Wed Apr 03 09:57:30 2002
Dumped from: Wed Apr 03 09:55:34 2002
Level 1 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Begin incremental restore
Initialize symbol table.
Extract directories from tape
Mark entries to be removed.
Calculate node updates.
Make node ./directory4
Make node ./directory5
Make node ./directory6
Find unreferenced names.
Remove old nodes (directories).
Extract new leaves.
Check pointing the restore
extract file ./file4
extract file ./file5
extract file ./file6
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
#
```

Alternative Steps

The following steps are an alternative to the previous Steps 5 and 6.

5. Restore the /export/home file system from the backup tapes. (This example uses an interactive, verbose restore to provide more detailed information.)

```
# ufsrestore iv /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Mon Jan 28 13:10:12 2002
Dumped from: the epoch
Level 0 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
```

```
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
     2 *./                   8  directory2        5  file2
     2 *../                  9  directory3        6  file3
     7  directory1           4  file1             3  lost+found/
```

The system lists files from the last Level 0 backup.

```
ufsrestore > add *
Warning: ./lost+found: File exists
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./file1
extract file ./file2
extract file ./file3
extract file ./directory1
extract file ./directory2
extract file ./directory3
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
Directories already exist, set modes anyway? [yn] n
ufsrestore > q
#
```

6. The information in the /etc/dumpdates file shows an incremental backup that was taken after the Level 0 backup. Load the next tape and perform the incremental restore.

```
# ufsrestore iv
Verify volume and initialize maps
Media block size is 64
Dump   date: Mon Jan 28 13:12:41 2002
Dumped from: Mon Jan 28 13:10:12 2002
Level 1 dump of /export/home on sys41:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
     2 *./             13  directory4      15  directory6      11  file5
     2 *../            14  directory5      10  file4           12  file6
```

```
ufsrestore > add *
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./file4
extract file ./file5
extract file ./file6
extract file ./directory4
extract file ./directory5
extract file ./directory6
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
ufsrestore > q
#
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Recovering Backup Files and File Systems (Level 1)

In this exercise, you read the backup tape from the previous exercise. You back up the / (root) file system, restore a single file from tape, and destroy and restore the / (root) file system.

Preparation

This exercise requires a system that is configured with a tape drive and a / (root) file system that is separate from the /usr and /var file systems. Identify the slice that holds the / (root) file system. From your instructor, get a tape appropriate for your system.

Tasks

Complete the following:

- Read the contents of both ufsdump files on the backup tape written in the previous exercise.

  (Steps 1–3 in Task 1 of the Level 2 lab)

- Reboot the system to run level S. Use the ufsdump command to create a backup tape of the / (root) file system on your system. Verify that the tape contains valid data for this file system. Allow the system to continue to boot to run level 3.

  (Steps 1–5 in Task 2 of the Level 2 lab)

- Use the ufsrestore i command to restore the /etc/inet/hosts file from tape, and place it below the /var/tmp directory.

  (Steps 1–6 in Task 3 of the Level 2 lab)

- Remove the /kernel, /platform, and /devices directories recursively. Abort the operating system, and attempt to boot the system from disk. Record what happens. Boot the system from the Solaris 9 Software 1 of 2 CD-ROM to run level S. Create a new file system on the / (root) slice. Use the ufsrestore command to reload the / (root) file system. Install a new boot block. Reboot the system, and eject the CD-ROM.

  (Steps 1–11 in Task 4 of the Level 2 lab)

Exercise: Recovering Backup Files and File Systems (Level 2)

In this exercise, you read the backup tape from the previous exercise. You back up the / (root) file system, restore a single file from tape, and destroy and restore the / (root) file system.

Preparation

This exercise requires a system that is configured with a tape drive and a / (root) file system that is separate from the /usr and /var file systems. Identify the slice that holds the / (root) file system. From the instructor, get a tape appropriate for your system.

Task Summary

In this exercise, you accomplish the following:

- Read the contents of both ufsdump files on the backup tape written in the previous exercise.
- Reboot the system to run level S. Use the ufsdump command to create a backup tape of the / (root) file system on your system. Verify that the tape contains valid data for this file system. Allow the system to continue to boot to run level 3.
- Use the ufsrestore i command to restore the /etc/inet/hosts file from tape, and place it below the /var/tmp directory.
- Remove the /kernel, /platform, and /devices directories recursively. Abort the operating system, and attempt to boot the system from disk. Record what happens. Boot the system from the Solaris 9 OE Software 1 of 2 CD-ROM to run level S. Create a new file system on the / (root) slice. Use the ufsrestore command to reload the / (root) file system. Install a new boot block. Reboot the system, and eject the CD-ROM.

Tasks

Complete the following tasks.

Task 1 – Read Your Previous Backup Tape

Complete the following steps:

1. Locate the backup tape written in the previous exercise, and load it into your tape drive.
2. Use the interactive restore command to view the content of the first Level 0 backup tape. Verify that the files are from the /export/home directory that you backed up. Enter q to quit the interactive restore.
3. Using a non-rewind device, move the tape to the next record, and view the contents of the second, incremental backup. Verify that the files you see are from the incremental backup. (The uucp directory is the one you added after the Level 0 backup.)

Task 2 – Create a Backup of the / (root) File System

Complete the following steps:

1. Log in as the root user, and open a terminal window. Shut down the system to run level 0. Then, boot the system to run level S. Supply the root password as required to enter run level S.
2. Verify that a tape is in your tape drive.
3. Use the ufsdump command to create a backup tape for the / (root) file system.
4. Verify that the / (root) file system is on the tape.
5. Allow the system to continue to boot to run level 3.

Task 3 – Restore the /etc/inet/hosts File From a Tape

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change to the /var/tmp directory.
2. Enter the ufsrestore i command to retrieve the /etc/inet/hosts file from the tape.
3. Change to the /etc/inet directory on the tape, and list the files in the directory.

4. Add the hosts file to the list of files to extract, and display the list.
5. Extract the hosts file from tape. Specify volume number 1. Do not set the owner and mode for ., and then quit the ufsrestore command.
6. Verify that the etc/inet/hosts file exists below the /var/tmp directory.

Task 4 – Destroy and Restore the / (root) File System

Complete the following steps:

1. Change to the / (root) directory, and remove the following critical system directories and their contents: /kernel, /platform, and /devices.
2. Press the Stop-A key sequence to abort the operating system. Attempt to boot the system from the boot disk.

   What happens?
3. Insert the Solaris 9 Software 1 of 2 CD-ROM. Boot the system from the CD-ROM to run level S.
4. Use the newfs command to create a new file system on the / (root) slice. (The slice should match the one you used earlier in the exercise when you created a backup of the / (root) file system.) Run the fsck command on the file system that you create.
5. Verify that your root backup tape is in the tape drive. Mount the new file system as the /a file system. Change to the /a directory.
6. Use the ufsrestore command to load the / (root) data into the new file system.
7. Remove the restoresymtable file.
8. Install a new boot block in Sectors 1 through 15 of the / (root) slice, by changing to the directory containing the boot block and entering the installboot command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
# installboot bootblk /dev/rdsk/c0t0d0s0
```

9. Change to the / (root) directory, and unmount the new file system.
10. Reboot the system.
11. Log in as the root user, and open a terminal window. Eject the Solaris 9 Software 1 of 2 CD-ROM.

Exercise: Recovering Backup Files and File Systems (Level 3)

In this exercise, you read the backup tape from the previous exercise. You back up the / (root) file system, restore a single file from tape, and destroy and restore the / (root) file system.

Preparation

This exercise requires a system that is configured with a tape drive and a / (root) file system that is separate from the /usr and /var file systems. Identify the slice that holds the / (root) file system. From the instructor get a tape appropriate for your system.

Task Summary

In this exercise, you accomplish the following:

- Read the contents of both ufsdump files on the backup tape written in the previous exercise.
- Reboot the system to run level S. Use the ufsdump command to create a backup tape of the / (root) file system on your system. Verify that the tape contains valid data for this file system. Allow the system to continue to boot to run level 3.
- Use the ufsrestore i command to restore the /etc/inet/hosts file from tape, and place it below the /var/tmp directory.
- Remove the /kernel, /platform, and /devices directories recursively. Abort the operating system, and attempt to boot the system from disk. Record what happens. Boot the system from the Solaris 9 Software 1 of 2 CD-ROM to run level S. Create a new file system on the / (root) slice. Use the ufsrestore command to reload the / (root) file system. Install a new boot block. Reboot the system, and eject the CD-ROM.

Tasks and Solutions

Complete the following tasks.

Task 1 – Read Your Previous Backup Tape

Complete the following steps:

1. Locate the backup tape written in the previous exercise, and load it into your tape drive.
2. Use the interactive restore command to view the contents of the first Level 0 backup tape. Verify that the files are from the /export/home directory that you backed up. Enter q to quit the interactive restore.

```
# ufsrestore iv
Verify volume and initialize maps
Media block size is 64
Dump   date: Fri Jan 25 08:38:53 2002
Dumped from: the epoch
Level 0 dump of /export/home on sys43:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
     2 *./              3712  default/          6  file3
     2 *../                4  file1             3  lost+found/
     7  core               5  file2
```

You should see the files from your /export/home directory.

```
ufsrestore > quit
```

3. Using a non-rewind device, move the tape to the next record, and view the contents of the second, incremental backup. Verify that the files you see are from the incremental backup. (The uucp directory is the one you added after the Level 0 backup.) Quit the interactive restore.

```
# mt -f /dev/rmt/0n fsf 1
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 64
Dump   date: Fri Jan 25 08:41:12 2002
Dumped from: Fri Jan 25 08:38:53 2002
Level 1 dump of /export/home on sys43:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore > ls
.:
     2 *./          2 *../       7424  uucp/
ufsrestore > q
#
```

Task 2 – Create a Backup of the / (root) File System

Complete the following steps:

1. Log in as the root user, and open a terminal window. Shut down the system to run level 0. Then boot the system to run level S. Supply the root password as required to enter run level S.

```
# init 0
(shutdown messages)
ok boot -s
```

2. Verify that a tape is in your tape drive.
3. Use the ufsdump command to create a backup tape for the / (root) file system.

```
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t0d0s0
```

4. Verify that the / (root) file system is on the tape.

```
# ufsrestore tvf /dev/rmt/0
```

The screen should scroll directory structures under / (root) first, followed by files.

5. Allow the system to continue to boot to run level 3.

```
# <Control-D>
```

Task 3 – Restore the /etc/inet/hosts File From a Tape

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change to the /var/tmp directory.

```
# cd /var/tmp
```

2. Enter the ufsrestore i command to retrieve the /etc/inet/hosts file from the tape.

```
# ufsrestore if /dev/rmt/0
ufsrestore > ls
```

You should see files and directories for the / (root) file system.

3. Change to the /etc/inet directory on the tape, and list the files in the directory.

```
ufsrestore > cd /etc/inet
ufsrestore > ls
```

You should see files and directories for the /etc/inet file system.

4. Add the hosts file to the list of files to extract, and display the list.

```
ufsrestore > add hosts
ufsrestore > marked
```

You should see the hosts file listed.

5. Extract the hosts file from tape. Specify volume number 1. Do not set the owner and mode for ., and then quit the ufsrestore command.

```
ufsrestore > extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] n
ufsrestore > q
```

6. Verify that the etc/inet/hosts file exists below the /var/tmp directory.

```
# ls etc/inet/hosts
etc/inet/hosts
```

Task 4 – Destroy and Restore the / (root) File System

Complete the following steps:

1. Change to the / (root) directory, and remove the following critical system directories and their contents: /kernel, /platform, and /devices.

```
# cd /
# rm -r /kernel /platform /devices
```

2. Press the Stop-A key sequence to abort the operating system. Attempt to boot the system from the boot disk.

```
ok boot
```

What happens?

The system fails to boot and displays the message:

```
Boot load failed.
The file just loaded does not appear to be executable
```

3. Insert the Solaris 9 Software 1 of 2 CD-ROM. Boot the system from the CD-ROM to run level S.

```
ok boot cdrom -s
```

4. Run the newfs command to create a new file system on the / (root) slice. (The slice should match the one you used earlier in the exercise when you created a backup of the / (root) file system.) Enter the fsck command on the file system that you create.

```
# newfs /dev/rdsk/c0t0d0s0
# fsck /dev/rdsk/c0t0d0s0
```

5. Verify that your root backup tape is in the tape drive. Mount the new file system as the /a file system. Change to the /a directory.

```
# mount /dev/dsk/c0t0d0s0 /a
# cd /a
```

6. Use the ufsrestore command to load the / (root) data into the new file system.

```
# ufsrestore rf /dev/rmt/0
```

7. Remove the restoresymtable file.

```
# rm restoresymtable
```

8. Install a new boot block in Sectors 1 through 15 of the / (root) slice, by changing to the directory containing the boot block and entering the installboot command.

```
# cd /usr/platform/`uname -m`/lib/fs/ufs
# installboot bootblk /dev/rdsk/c0t0d0s0
```

9. Change to the / (root) directory, and unmount the new file system.

```
# cd /
# umount /a
```

10. Reboot the system.

```
# init 6
```

11. Log in as the root user, and open a terminal window. Eject the Solaris 9 Software 1 of 2 CD-ROM.

```
# eject cdrom
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Introducing Disaster Recovery Fundamentals

This section introduces some basic concepts and criteria for the planning and anticipation of a disaster. Understanding what disaster recovery is should be the first step you take toward defining and implementing a disaster recovery plan.

Identifying the fundamentals of a disaster recovery plan start with defining what a disaster is. In most cases, it can be defined as follows: any unplanned, extended loss of critical business applications due to a lack of computer processing capabilities. In this definition, extended is defined by the individual business at hand. Some businesses might suffer severe losses in one hour while other businesses might not suffer severe losses.

Disaster Scenarios That Can Result in a Loss of Data

Disasters, both natural and man-made, happen every day. For example, flooding in Chicago might disrupt operations in 400 data centers in Chicago or all over the United States, or a hurricane could disable the reservation system of a major airline. A major storm could interrupt power to a geographic area for an extended period of time. A fire could destroy business critical infrastructure.

Many scenarios could involve the loss of major components of a business-critical system or systems. You have many reasons to consider having a well-thought-out disaster recovery plan in place. The most important reasons might be the ones you cannot think of.

Disaster Recovery Plan

No single document can present a comprehensive analysis of how to prepare a disaster recovery plan. However, this section describes some of the key steps to create a plan.

Solicit Department Input

Gather input from the departments within your company. Individuals in these departments know the type of data they create and the importance of the data. If you work at a large company, you might organize a planning team so that the team can inform management on the status of the disaster recover program. Departments might be tasked to complete periodic tests to verify implementation of the disaster recovery program. The departments can present any findings of gaps or risks they encounter.

Acquire Management Approval

Involve management early in the decision-making process. This is an important step if you are to obtain the necessary resources and time required from each area of your organization.

You, or your planning team, should complete a study of the disaster recovery plan and include an estimate of the cost of a disaster, as well as an estimate of the possible costs and time to implement a disaster recovery strategy. When management understands the financial, physical, and business costs associated with a disaster, the planning team is able to build a strategy and ensure that the strategy is implemented across the organization.

Develop a Budget

Cost-justifying a disaster recovery program is relatively simple. No competitive organization can afford to be without a comprehensive program. In developing a budget for your program, you should consider the following:

- Your current cost of downtime

  Look at both the total cost per minute as well as the cost per event, and include the intangible or *soft-dollar* costs, such as loss of productivity and diminished customer confidence.

- The cost of re-creating data

  Consider the time lost re-creating files, the expense of retrieving data from crashed hard disks, the cost of unavailable data when you need it, and the additional costs involved in re-creating lost data when there is no backup copy available.

- The cost of expert assistance

  Compared to the costs of downtime, the costs of hiring experts to assist with the disaster recovery program are minimal.

Evaluating the Critical Factors for the Plan

Creating the actual disaster recovery plan varies from one company to another. Every company must first evaluate the following critical factors for a disaster recovery plan:

- What is the greatest risk?

  Is your company most susceptible to natural events, such as earthquakes, fires, or floods; mechanical failures, such as hardware and software problems; human error; or intrusions, such as hacker attacks and viruses?

- How are various groups or departments within your company affected by downtime?
  - For each group, how vital is access to data?
  - How long could each group or department function without access to data?
  - Is the hardware and data all centrally located, or are there alternative sites or departments that can provide the resources lost in the event of a disaster?
- What preventative measures are in place right now?
  - Is there a disaster team?
  - Have you defined a backup strategy?
  - Where is the most valuable data stored, and is it adequately protected?
  - Have you documented inventory with schematics, specifications, passwords, menus, utilities, and startup files?
  - Do your facilities have backup data lines and connections?
- How can you recover your data?
  - Who is in charge of managing this process?
  - Is there a communication procedure?
  - Could you recover your data at a different facility or geographical area, or with different personnel?
  - Do you have the need for a hotsite? If so, do you establish the hotsite or use a hotsite vendor?
  - How long would it take to fix your existing facility under various disaster scenarios?

- Should you out-source the data recovery process to a third-party company?
- Do you have emergency contacts with your suppliers?
- What are the approval procedures for emergency planning?
- What happens if you cannot reach key personnel?

Create a Procedure

After you have analyzed the previous choices, you are ready to establis the actual procedures that you must follow in the event of a disaster. The procedures must:

- Define how to andle various aspects of the network, including loss of servers, bridges and routers, communications links, and so on
- Specify who arranges for repairs or reconstruction and how the data recovery process occurs
- Include a checklist or test procedure to verify that everything works when the repairs and data recovery have taken place

Test the Procedure

You must test the plan—not just once, but often. You should determine how frequently you test the plan, by considering the following factors:

- Personnel changes
- System changes
- Network changes

You must measure the success of each test. You must define the objective measurements to verify that your plan is effective.

Keep Pace With Changes

Change is the only constant in the corporate world, and it is important that your disaster recovery plan takes into account the impact of change. Consider the following:

- Are there processes in place to include new departments and facilities in the disaster recovery plan?
- Is there a regularly scheduled review of the plan?
- Who maintains the disaster management team roster?
- Could someone new execute the plan?
- How do you communicate changes or modifications to the plan to people who are affected by the changes?

Importance of Off-Site Backups

Having a comprehensive and quality backup program is vital to a good disaster recovery plan. The value of backed-up data depends on the security and physical protection of that media.

Many scenarios can account for data loss or corruption. These scenarios fall short of what is typically considered a disaster. In some instances, you might need to recover data lost due to human error or a minor hardware failure. To recover from data corruption you need to gain quick access to backup media, such as tape.

Many Information Technology (IT) providers keep a local copy of their backups and send a tested copy of the same backups to an off-site storage service provider. Off-site storage gives you the opportunity to keep an integrity-tested copy of all your data in a safe location away from the site of business. Off-site providers store your backups in a disaster-ready environment. These sites usually are built to protect your backups from most known disaster scenarios, such as fire, flood, theft, and so on. Without an off-site storage solution, you could place even the most comprehensive disaster recovery program at great risk.

Components Required to Operate a Hotsite

You might decide that there is sufficient need and business justification to implement a hotsite as part of your disaster recovery plan. A hotsite is a virtual replication of your critical business computer operations. A hotsite is also known as a disaster recovery site.

If the cost of establishing a hotsite is too high, you can use commercial hotsite vendors. Some disaster recovery vendors offer a variety of options. In a hotsite environment you might get pre-installed computers, networking equipment, telecommunications equipment, raised flooring, air conditioning, technical support, and uninterruptable power supplies.

Importance of Disaster Recovery Drills

As part of a comprehensive disaster recovery plan, testing is one of the most critical steps. After successfully testing your disaster recovery plan, you must continue to test your plan on a regular basis through disaster recovery drills.

Conducting disaster recovery drills allows you the opportunity to adjust and update your disaster recovery plan as needed. You have the opportunity to account for personnel changes, system changes, application changes, and so on.

Conducting disaster recovery drills on a regular basis also keeps the importance of disaster recovery as a priority in your every day computer operations. The more you practice for disaster recovery scenarios, the quicker you are able to recover, saving more of your customer base and your business earnings.

Module 17

Backing Up a Mounted File System With a UFS Snapshot

Objectives

Upon completion of this module, you should be able to:

- Create a UFS snapshot
- Back up the snapshot file

The following course map shows how this module fits into the current instructional goal.

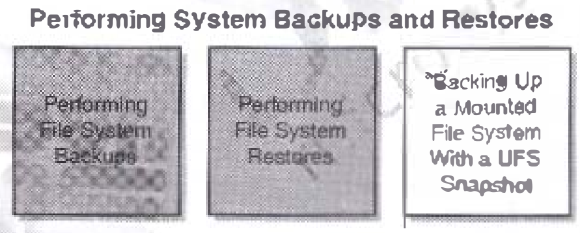

Figure 17-1 Course Map

Creating a UFS Snapshot

The UFS Copy on Write Snapshots feature provides administrators of non-enterprise-level systems an online backup solution for ufs file systems. This utility enables you to create a point-in-time copy of a ufs file system, called a snapshot, as an online backup. You can create the backup while the file system is mounted and the system is in multiuser mode.

Note – The UFS snapshots are similar to the Instant Image product. Instant Image allocates space equal to the size of the entire file system that is being captured. However, the file system data saved by UFS snapshots occupies only as much disk space as needed.

Using the `fssnap` Command

You use the `fssnap` command to create, query, or delete temporary read-only snapshots of ufs file systems.

The format for the `fssnap` command is:

```
/usr/sbin/fssnap -F FSType -V -o special_option(s) mount-point | special
```

Table 17-1 shows some of the options for the `fssnap` command.

Table 17-1 Options for the fssnap Command

Option	Description
-d	Deletes the snapshot associated with the given file system. If the -o unlink option was used when you built the snapshot, the backing-store file is deleted together with the snapshot. Otherwise, the backing-store file (which contains file system data) occupies disk space until you delete it manually.
-F *FSType*	Specifies the file system type to be used.
-i	Displays the state of an *FSType* snapshot.
-V	Echoes the complete command line but does not execute the command.
-o	Enables you to use *special_options*, such as the location and size of the backing-store (bs) file.

To create a UFS snapshot, specify a backing-store path and the actual file system to be captured. The following is the command format:

```
# fssnap -F ufs -o bs=backing_store_path /file-system
```

Note – The `backing_store_path` can be a raw device, the name of an existing directory, or the name of a file that does not already exist.

The following example uses the `fssnap` command to create a snapshot of the `/export/home` file system.

```
# fssnap -F ufs -o bs=/var/tmp /export/home
/dev/fssnap/0
```

The snapshot subsystem saves file system data in a file called a backing-store file before the data is overwritten. Some important aspects of a backing-store file are:

- A backing-store file is a bit-mapped file that takes up disk space until you delete the UFS snapshot.
- The size of the backing-store file varies with the amount of activity on the file system being captured.
- The destination path that you specify on the `fssnap` command line must have enough free space to hold the backing-store file.
- The location of the backing-store file must be different from that of the file system you want to capture in a UFS snapshot.
- A backing-store file can reside on different types of file systems, including another `ufs` file system or a mounted `nfs` file system.

The `fssnap` command creates the backing-store file and two read-only virtual devices. The block virtual device, `/dev/fssnap/0`, can be mounted as a read-only file system. The raw virtual device, `/dev/rfssnap/0`, can be used for raw read-only access to a file system.

These virtual devices can be backed up with any of the existing Solaris OE backup commands. The backup created from a virtual device is a backup of the original file system when the UFS snapshot was taken.

Note – When a UFS snapshot is first created, the file system locks temporarily. Users might notice a slight pause when writing to this file system. The length of the pause increases with the size of the file system. There is no performance impact when users are reading from the file system.

Limiting the Size of the Backing-Store File

Before creating a UFS snapshot, use the `df -k` command to check that the backing-store file has enough disk space to grow. The size of the backing-store file depends on how much data has changed since the previous snapshot was taken.

You can limit the size of the backing-store file by using the `-o maxsize=n` option of the `fssnap` command, where *n* (k, m, or g) is the maximum size of the backing-store file specified in Kbytes, Mbytes, or Gbytes.

Additionally, you can place a minimum size on the backing-store file by using the `-o minsize=n` option with the `fssnap` command.

Caution – If the backing-store file runs out of disk space, the system automatically deletes the UFS snapshot, which causes the backup to fail. The active `ufs` file system is not affected. Check the `/var/adm/messages` file for possible UFS snapshot errors.

Note – You can force an unmount of an active `ufs` file system, for which a snapshot exists (for example, with the `umount -f` command). This action deletes the appropriate snapshot automatically.

The following example creates a snapshot of the `/export/home` file system, and limits the backing-store file to 500 Mbytes.

```
# fssnap -F ufs -o bs=/var/tmp,maxsize=500m /export/home
/dev/fssnap/0
```

Displaying Information for a ufs File System Snapshot

You can use either fssnap command to display UFS snapshot information.

The following example displays a list of all the current UFS snapshots on the system. The list also displays the corresponding virtual device for each snapshot.

```
# fssnap -i
0     /export/home
1     /usr
2     /database
```

You use the -i option to the /usr/lib/fs/ufs/fssnap command to display detailed information for a specific UFS snapshot that was created by the fssnap command.

The following example shows the details for the /export/home snapshot.

```
# /usr/lib/fs/ufs/fssnap -i /export/home
Snapshot number                 : 0
Block Device                    : /dev/fssnap/0
Raw Device                      : /dev/rfssnap/0
Mount point                     : /export/home
Device state                    : idle
Backing store path              : /var/tmp/snapshot0
Backing store size              : 0 KB
Maximum backing store size      : 512000 KB
Snapshot create time            : Mon Apr 22 08:58:33 2002
Copy-on-write granularity       : 32 KB
```

Backing Up the UFS Snapshot File

The virtual devices that contain the UFS snapshot act as standard read-only devices, which enable you to back up the virtual device in the same manner as you would back up a file system.

Performing a Backup of a UFS Snapshot

You can use the `tar` command or the `ufsdump` command to back up a UFS snapshot.

Using the `tar` Command to Back Up a Snapshot File

If you use the tar command to back up the UFS snapshot, mount the snapshot before backing it up. The following procedure demonstrates how to do this type of mount.

1. Create the mount point for the block virtual device.

```
# mkdir -p /backups/home.bkup
```

2. Mount the block virtual device to the mount point.

```
# mount -F ufs -o ro /dev/fssnap/0 /backups/home.bkup
```

3. Change directory to the mount point.

```
# cd /backups/home.bkup
```

4. Use the tar command to write the data to tape.

```
# tar cvf /dev/rmt/0 .
```

Using the `ufsdump` Command

If you use the ufsdump command to back up a UFS snapshot, you can specify the raw virtual device during the backup.

```
# ufsdump 0uf /dev/rmt/0 /dev/rfssnap/0
```

Verify that the UFS snapshot is backed up.

```
# ufsrestore tf /dev/rmt/0
```

Performing an Incremental Backup of a UFS Snapshot

Incremental snapshots contain files that have been modified since the last UFS snapshot. You use the `ufsdump` command with the N option to create an incremental UFS snapshot, which writes the name of the device being backed up, rather than the name of the snapshot device to the `/etc/dumpdates` file.

The following example shows how to use the `ufsdump` command to create an incremental backup of a file system.

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsk/c1t0d0s0 /dev/rfssnap/0
```

Next you would verify that the UFS snapshot is backed up to tape.

```
# ufsrestore tf /dev/rmt/0
```

To understand incremental backups of snapshots, consider the following demonstration:

1. Create a snapshot of the `/extra` file system that is going to be backed up while the file system is mounted.

```
# fssnap -o bs=/var/tmp /extra
/dev/fssnap/0
#
```

2. Verify that the snapshot was successful, and view detailed information about the snapshot.

```
# fssnap -i
   0    /extra
# /usr/lib/fs/ufs/fssnap -i /extra
Snapshot number               : 0
Block Device                  : /dev/fssnap/0
Raw Device                    : /dev/rfssnap/0
Mount point                   : /extra
Device state                  : idle
Backing store path            : /var/tmp/snapshot0
Backing store size            : 0 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Thu Apr 04 10:34:21 2002
Copy-on-write granularity     : 32 KB
```

3. Make a directory that will be used to mount and view the snapshot data.

```
# mkdir /extrasnap
#
```

4. Mount the snapshot to the new mount point, and compare the size of the file system and the snapshot device.

```
# mount -o ro /dev/fssnap/0 /extrasnap
# df -k |grep extra
/dev/dsk/c1t0d0s0      1294023        9 1242254     1%    /extra
/dev/fssnap/0          1294023        9 1242254     1%    /extrasnap
```

5. Edit a file under the /extra directory and make it larger, and then compare the size of the file system and the snapshot device.

```
# vi file1
(yank and put text, or read text in from another file)
# df -k |grep extra
/dev/dsk/c1t0d0s0      1294023       20 1242243     1%    /extra
/dev/fssnap/0          1294023        9 1242254     1%    /extrasnap
```

Observe that the file system grew in size while the snapshot file did not.

6. Perform a full backup with the N option of the ufsdump command.

```
# ufsdump 0ufN /dev/rmt/0 /dev/rdsk/c1t0d0s0 /dev/rfssnap/0
  DUMP: Writing 32 Kilobyte records
  DUMP: Date of this level 0 dump: Thu 04 Apr 2002 10:49:38 AM MST
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rfssnap/0 (sys41:/extrasnap) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 262 blocks (131KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 254 blocks (127KB) on 1 volume at 1814 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Thu 04 Apr 2002 10:49:38 AM MST
```

7. Verify the backup.

```
# ufsrestore tf /dev/rmt/0
         2      .
         3      ./file1
         4      ./file2
         5      ./file3
         6      ./file4
#
```

8. Unmount the back up device and remove the snapshot.

```
# umount /extrasnap
# fssnap -d /extra
# rm /var/tmp/snapshot0
#
```

9. Make some changes to the /extra file system, such as copying some files, and then re-create the snapshot.

```
# cp file1 file5
# cp file1 file6
# fssnap -o bs=/var/tmp /extra
/dev/fssnap/0
#
```

10. Re-mount the snapshot device, and compare the size of the file system and the snapshot device.

```
# mount -o ro /dev/fssnap/0 /extrasnap
# df -k |grep extra
/dev/dsk/c1t0d0s0      1294023      46 1242217      1%      /extra
/dev/fssnap/0          1294023      46 1242217      1%      /extrasnap
#
```

11. Perform an incremental backup with the N option of the ufsdump command.

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsk/c1t0d0s0 /dev/rfssnap/0
  DUMP: Writing 32 Kilobyte records
  DUMP: Date of this level 1 dump: Thu 04 Apr 2002 10:59:11 AM MST
  DUMP: Date of last level 0 dump: Thu 04 Apr 2002 10:49:38 AM MST
  DUMP: Dumping /dev/rfssnap/0 (sys41:/extrasnap) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 294 blocks (147KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 254 blocks (127KB) on 1 volume at 1693 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 1 dump on Thu 04 Apr 2002 10:59:11 AM MST
#
```

12. Verify the backup.

```
# ufsrestore tf /dev/rmt/0
         2      .
         7      ./file5
         8      ./file6
#
```

Notice that the backup of the snapshot contains only the files that were added since the previous Level 0 backup.

Restoring Data From a UFS Snapshot Backup

The backup created from a virtual device is a backup of the original file system when the UFS snapshot was taken.

You restore a UFS snapshot from a backup tape in the same manner as you would the backup of an original file system.

To restore the demo directory from the snapshot backup of the /usr file system, complete the following steps:

1. Load the tape that contains the snapshot backup of the /usr file system into the tape drive.
2. Change to the /usr file system.

```
# cd /usr
```

3. Perform the a ufsrestore command.

```
# ufsrestore if /dev/rmt/0
ufsrestore > add demo
ufsrestore > extract
Specify next volume #: 1
set owner/mode for '.'? [yn] n
ufsrestore > quit
```

4. Verify that the demo directory exists, and eject the tape.

Deleting a UFS Snapshot

Deleting a UFS snapshot from the system is a multistep process and order-dependant. First, unmount the snapshot device, and then delete the snapshot. Finally, remove the backing-store file.

```
# umount /dev/fssnap/0
# fssnap -d /export/home
# rm /backing_store_file
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Working With UFS Snapshots (Level 1)

In this exercise, you create a UFS snapshot of the /opt file system, display detailed information for the UFS snapshot, and then remove the snapshot and backing-store file.

Tasks

Complete the following tasks:

- Create a snapshot of the /opt file system
- View the contents of the backing-store directory
- Display detailed information about the snapshot
- Remove the snapshot from the system

Exercise: Working With UFS Snapshots (Level 2)

In this exercise, you create a UFS snapshot of the /opt file system, display detailed information for the UFS snapshot, and then remove the snapshot and backing-store file.

Task Summary

In this exercise, you accomplish the following:

- Create a snapshot of the /opt file system
- View the contents of the backing-store directory
- Display detailed information about the snapshot
- Remove the snapshot from the system

Tasks

Complete the following steps:

1. Create a snapshot of the /opt file system without specifying a file name for the backing-store file.
2. View the contents of the /var/tmp file system.

   What is the default name assigned to a backing-store file?
3. Display the detailed information about the snapshot.

   What is the maximum backing-store file size for the snapshot?
4. Delete the snapshot from the system.
5. View the contents of the /var/tmp file system.

   Has the backing-store file been removed?
6. Remove the backing-store file that you created in Step 1.

Exercise: Working With UFS Snapshots (Level 3)

In this exercise, you create a UFS snapshot of the /opt file system, display detailed information for the UFS snapshot, and then remove the snapshot and backing-store file.

Task Summary

In this exercise, you accomplish the following:

- Create a snapshot of the /opt file system
- View the contents of the backing-store directory
- Display detailed information about the snapshot
- Remove the snapshot from the system

Tasks and Solutions

Complete the following steps:

1. Create a snapshot of the /opt file system without specifying a file name for the backing-store file.

```
# fssnap -F ufs -o bs=/var/tmp /opt
```

2. View the contents of the /var/tmp file system.

   What is the default name assigned to a backing-store file?

   snapshot0

3. Display the detailed information about the snapshot.

```
# /usr/lib/fs/ufs/fssnap -i /opt
```

   What is the maximum backing-store file size for the snapshot?

   Unlimited

4. Delete the snapshot from the system.

```
# fssnap -d /opt
```

5. View the contents of the /var/tmp file system. Has the backing-store file been removed?

   No

6. Remove the backing-store file you created in Step 1.

```
# rm /var/tmp/snapshot0
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Index

C

D

L

M

N

O

P

S

V

W

Z